UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE ESTUDIOS

Título

Diseño y desarrollo de un SGA para la empresa Norpoo Prototipos S.L

Autor/es

JORGE BELLA SÁENZ DE INESTRILLAS

Director/es

CÉSAR DOMÍNGUEZ PÉREZ

Facultad

Facultad de Ciencia y Tecnología

Titulación

Grado en Ingeniería Informática

Departamento

MATEMÁTICAS Y COMPUTACIÓN

Curso académico

2019-20

publicaciones.unirioja.es
E-mail: publicaciones@unirioja.es

UNIVERSIDAD DE LA RIOJA

Facultad de Ciencia y Tecnología

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Diseño y desarrollo de un SGA para la empresa Norpoo Prototipos

Realizado por:

Jorge Bella Sáenz de Inestrillas

Tutelado por:

César Domínguez Pérez

Logroño, Junio, 2020

Índice

Resumen en Español 4
Resumen en Inglés 4
1. Análisis y planificación del proyecto 5
1.1. Contexto de la empresa 5
1.2. Situación inicial del problema 5
1.3. Acciones previas para atacar el problema 6
1.3.1. Identificación de las estanterías 6
1.3.2. Volcado de la información disponible en un fichero 7
1.4. ¿Qué es lo que se quiere realizar en este trabajo? 8
1.5. Requisitos del proyecto 9
1.5.1. Requisitos de datos 9
1.5.2. Requisitos funcionales 10
1.6. Herramientas de trabajo 11
1.6.1. Sistema Gestor de Base de Datos 11
1.6.2. Entorno de desarrollo de la aplicación 12
1.7. Entregables 12
1.8. Diagrama EDT 13
1.9. Planificación temporal 14
1.9.1. Descripción de actividades 14
1.9.2. Cronograma 15
1.9.3. Dependencias 16
1.9.4. Hitos 16
1.9.5. Estimaciones de dedicación 18
1.10. Recursos materiales 19
1.11. Comunicaciones 19
1.11.1. Medios de comunicación 19
1.11.2. Comunicación interna 19
1.12. Riesgos 19
1.13. Partes interesadas 21
2. Diseño 21
2.1. Diseño de la base de datos 21
2.1.1. Requisitos de datos 21
2.1.2. Diagrama Entidad/Relación 21
2.1.3. Diseño Lógico/Transformación a Modelo Relacional 22

2.1.4. Normalización ... 22
2.1.5. Diseño físico ... 22
2.2. Diseño de la Aplicación ... 22
2.2.1. Diagrama de casos de uso ... 22
2.2.2. Especificación de los casos de uso ... 23
2.2.3. Diagrama de Clases de la aplicación ... 24
2.2.4. Diagramas de actividad ... 25
2.2.5. Diseño de la interfaz gráfica ... 27
3. Implementación ... 28
3.1. Implementación de la base de datos ... 28
3.1.1. Creación de la base de datos ... 28
3.1.1.1. Creación de las tablas ... 29
3.1.1.2. Restricciones de tablas ... 30
3.1.1.3. Creación de usuarios ... 31
3.1.1.4. Inserción de los datos ... 32
3.2. Implementación de la aplicación ... 33
3.2.1. Implementación del modelo de datos ... 33
3.2.1.1. Clase Producto ... 33
3.2.1.2. Clase ProductoES ... 34
3.2.1.3. Clase ProductoFS ... 34
3.2.1.4. Clase Ubicación ... 35
3.2.2. Implementación de la capa de Persistencia ... 35
3.2.2.1. Clase GestorBD ... 36
3.2.3. Implementación de la Lógica de Negocio ... 39
3.2.3.1. Clase LogicaNegocio ... 39
3.2.4. Implementación de la capa de Presentación ... 40
3.2.4.1. Clase Interfaz ... 40
3.2.4.1.1. Parte gráfica ... 40
3.2.4.1.2. Parte de desarrollo del código ... 41
4. Etiquetado de productos ... 42
5. Consideraciones sobre el despliegue ... 45
6. Otros aspectos y conclusiones ... 46
6.1. Manual de usuario ... 46
6.2. Mantenimiento ... 46
6.3. Otras conclusiones ... 46

7. Bibliografía 46

Resumen en Español

Para cumplir con los estándares de calidad de la norma ISO 9001:2015, la empresa Norpoo Prototipos S.L necesita mejorar la trazabilidad y el control del inventario del material almacenado.

Por esta razón se trabajará y se realizará un Sistema de Gestión del Almacén que cubra aquellas necesidades que permitan conseguir esa certificación correctamente.

Este sistema consistirá en una aplicación de escritorio desarrollada en Java, conectada a una base de datos situada en una instancia SQL Server, donde se encontrará la información necesaria de todos los productos almacenados en la empresa.

Además, gracias a la adquisición reciente de una impresora de etiquetas, podremos conectar esa base de datos en SQL Server al software de la impresora para darle al componente guardado un sello de identidad propio de la empresa.

Resumen en Inglés

To comply with the quality standards of ISO 9001:2015, the company Norpoo Prototipos S.L needs to improve the traceability and inventory control of the stored material.

For this reason, a Warehouse Management System will be worked on and carried out to cover those needs that allow achieving this certification correctly.

This system will consist of a desktop application developed in Java, connected to a database located in a SQL Server instance, where you will find the necessary information on all the products stored in the company.

In addition, thanks to the recent acquisition of a label printer, we will be able to connect that database in SQL Server to the printer software to give the saved component a company identity seal.

1. Análisis y planificación del proyecto

1.1. Contexto de la empresa

Norpoo Prototipos S.L. es una empresa que se dedica al desarrollo y a la fabricación de tarjetas electrónicas, así como al diseño, industrialización de los equipos, realización de test de validación de piezas, montajes, etc.

Actualmente tienen mucho volumen de entrada de componentes dado el aumento de proyectos. Esto es debido a la certificación ISO:9001 conseguida en Diciembre de 2018, por ello, el guardado de las diferentes piezas para su localización, uso posterior y trazabilidad de recursos, así como la rotación del stock en producción es algo fundamental para conservar la normalización conseguida anteriormente y el buen funcionamiento de la empresa.

1.2. Situación inicial del problema

Antes de comenzar a analizar el problema a solucionar, se ha evaluado cuál es la situación actual del almacén como de la red informática interna de la empresa, y se han llegado a las siguientes conclusiones:

- Sobre el almacén:
    - En las estanterías que conforman el almacén, se encuentran productos con referencias repetidas, esto tiene sentido cuando hay varios productos iguales, pero no se respeta el criterio FIFO para dar salida a estos productos. Es decir, que no siempre el producto que más tiempo lleva en el almacén es el que se usa primero, por tanto, no se tiene en cuenta el albarán del pedido del producto, que es identificativo para saber qué producto es más antiguo.
    - No se controla ni la trazabilidad ni el control del stock disponible en la empresa de una manera correcta. Se sigue correctamente desde su entrada en la empresa hasta la colocación del producto en el almacén; pero, cuando el producto está siendo usado, no se contabiliza el número de ellos que se está usando. Además, no se controla el stock disponible en caso de que ese producto en uso se acabe. Ahora es viable controlarlo de manera personal, debido a que el almacén es pequeño, pero puede ser un problema creciente en cuanto la empresa crezca más.
    - El etiquetado utilizado es el que el proveedor envía en la remesa de productos pedidos, por tanto hay etiquetas de todo tipo y cada una tiene diferentes campos. Por tanto, la información del producto es difícil de comprender debido a la multitud de etiquetas. Para ello normalizaremos el etiquetado creando un diseño propio.

- Sobre la infraestructura:
    - Se posee una red pequeña de 4 ordenadores, 2 en la oficina y 2 en la zona de trabajo. En esa zona de trabajo es donde se encuentra el almacén. Todos estos ordenadores están conectados a la misma red local, pero cada uno tiene una función específica dentro de la empresa. Pero, a simple vista no se observan las diferencias de jerarquía que hay en los equipos. Por esta razón debemos preparar los equipos para el despliegue final de la aplicación, aportando niveles de jerarquía y seguridad a la estructura informática.

1.3. Acciones previas para atacar el problema

Después de analizar el problema y las posibles acciones a mejorar se han realizado las siguientes acciones previas para comenzar el desarrollo del proyecto:

1.3.1. Identificación de las estanterías

En primer lugar se han identificado las estanterías donde se almacenan los componentes con códigos de letras y números, por ejemplo:

- Cada estantería se le nombra con una letra, empezando por la A, siguiendo con B, C…
- A cada piso de la estantería se le asigna un número, siendo el número 1 el piso más alto y aumentando en una unidad conforme baje el piso de almacenamiento del componente.
- A cada fila también se le asigna un número, siendo 1 la fila que se encuentra más a la izquierda de la estantería y aumentando en 1 por cada fila que haya en ella.
- Con las columnas de la estantería procedemos al igual que con las filas.

En la ilustración 1 se muestra un ejemplo de estantería del almacén:

Ilustración 1. Tres de las estanterías que conforman el almacén de la empresa.

Se puede observar la codificación de las estanterías en la imagen, siendo sencillo de ubicar un componente, por ejemplo:

- Las gavetas azules que están en el centro de la imagen (aquellas que están solas en un piso), se ubican en la estantería E, piso 2, fila 1 y 2 respectivamente, y columna 1.

Algunos componentes electrónicos deben almacenarse en unas condiciones especiales de temperatura y humedad. Esto sucede debido a que son productos muy sensibles y pueden estropearse si se guardan en las condiciones normales de temperatura que hay en la empresa.

Para depositar estos componentes, se posee una cámara especial que simula las condiciones ideales de preservación de estos productos. También entran en el proceso de la Gestión del Almacén y se identificará tanto la cámara, como los pisos que tenga para el almacenado de componentes.

El nombre de la cámara para su identificación será AD101, que es el modelo del dispositivo.

Ilustración 2. Cámara de componentes sensibles Mekko AD101

1.3.2. Volcado de la información disponible en un fichero

Con todas las referencias disponibles en el almacén se ha creado un fichero en formato Excel, conteniendo los siguientes campos:

- Referencia del fabricante del producto
- Proveedor del producto
- Albarán del pedido del producto
- Una descripción breve del componente
- Si el producto necesita ser guardado en la cámara especial o no
- Estantería
- Piso
- Fila
- Columna

En la ilustración 3 se muestra una porción del fichero creado.

REF. FABRIC.	PROVEEDOR	ALBARAN	DESCRIPCION	HUM	Estanteria	Piso	Fila	Columna
C1206C221G5GAC7800	AB Electronic	Dummie	C 220pF 50V 2%	No				
SS1P3L-M3/84A	Digi-Key	62784601	Diodo Schottky (SMP)	No	D	2	1	1
LZ1-00R202-0000	Digi-Key	62784601	LED Rojo	No	D	2	2	1
SRE6603-103M	Digi-Key	62784601	Bob. 10mH	No	B	4	1	1
LZ1-00B202-0000	Digi-Key	62784601	LED Azul	No	D	2	2	2
LZ1-00G102-0000	Digi-Key	62784601	LED Verde	No	D	2	2	3
C1206C476M9PACTU	Digi-Key	62784601	C 47uF 6.3V 20% 1206	No	E	3	1	1
ERA-3AEB102V	Digi-Key	62784601	R 1K 0.1% 0603	No	F	2	1	1
CG0603MLC-05LE	Digi-Key	62784601	TVS Disp. Tª 0603	No	I	4	4	1
UMK107BJ105KA-T	Digi-Key	62784601	C 1uF 50V 10% 0603	No	E	3	1	2
CC0603JRNPO9BN100	Digi-Key	62784601	C 10pF 50V 5% 0603	No	E	3	1	3
RC0603FR-07100KL	Mouser	48107188	R 100K 1% 0603	No	F	2	1	2
RC0603JR-07220RL	Digi-Key	62784601	R 220 5% 0603	No	F	2	1	3
ERJ-3EKF4701V	Digi-Key	62784601	R 4K7 1% 0603	No	F	2	1	4
ERT-J1VG103FA	Digi-Key	62784601	R 10K 1% 0603	No	F	2	1	5
06031C103K4Z2A	Digi-Key	62784601	C 10nF 100V 10% 0603	No	E	3	1	4
BCX5616TA	Mouser	48107188	Diodo SOT-89-3	No	D	2	2	4
U.FL-R-SMT(01)	Farnell	DLG17263381	Antena	No	C	4	1	1
MCP111T-195I/TT	Mouser	48107188	CI Open Drain SOT23-3	No	G	2	2	1
LM2734YMK/NOPB	Farnell	DGB12700198	CI Reg.conmutación SOT23-6	No	G	2	2	1
SRR0604-2R5ML	RS	Justo Contreras	Bob. 2.5uF	No	B	4	1	1
MCP73833-AMI/UN	Digi-Key	63039698	CI Lithium Ion 10-MSOP	No	G	2	2	1
MCP73833-AMI/UN	Digi-Key	64027827	CI Lithium Ion 10-MSOP	No	G	2	2	1
MCWR06X3162FTL	Farnell	DGB12735235	R 31K6 1% 0603	No	F	2	4	2
878580002	Digi-Key	62784601	Conector	No	H	3	1	1
M20-8890845	Mouser	48107188	Conector	No	H	5	1	
LMR12010XMKE/NOPB	Mouser	48107188	CI Reg. Comunicación TSOT23-6	No	G	2	2	1
KPJA-2107QBC-D	Farnell	DGB12735235	LED Azul	Si	1			
KPJA-2107SURCK	Farnell	DGB12735235	LED Rojo	Si	1			
KPJA-2107SGC	Farnell	DGB12735235	LED Verde	Si	1			
M22-3010200	Digi-Key	62942002	Conector	No	H	3	1	1

Ilustración 3. Fichero Excel orientativo para el almacén

Tras introducir los datos en la tabla Excel surgen los siguientes problemas:

- No se conoce el stock disponible de los productos almacenados en inventario con certeza, o no está almacenado previamente. Esto supone un problema gravísimo para conocer la trazabilidad y el inventario completo que hay en la empresa.
- Existen productos en el fichero que su cantidad es cero, por tanto no deben estar en el almacén. Aunque se desee almacenar un histórico de productos que han podido estar en las estanterías del almacén, aquellos productos “vacíos” no deben mezclarse con los componentes actuales que conforman las estanterías de la empresa.

1.4. ¿Qué es lo que se quiere realizar en este trabajo?

Con todas las variables analizadas, tanto en la situación actual del problema, como en el análisis y las acciones previas para solucionarlo, se ha definido que el proyecto consistirá en lo siguiente:

- Se diseñará, implementará y desplegará una aplicación de escritorio sencilla y funcional para el control del almacén de la empresa
- Esta aplicación será denominada como SGA (Sistema de Gestión de Almacén) y constará de una interfaz gráfica intuitiva, donde su funcionalidad se basará en la manipulación de la base de datos de la información de los componentes almacenados.
- Contendrá las operaciones básicas y funcionales de manipulación y obtención de información de la base de datos, con algunas particularidades en alguna de estas opciones. Serán explicadas en los requisitos funcionales del proyecto.
- Para conseguir la jerarquía de usuarios, cada equipo estará protegido con el sistema de contraseñas de Windows, y para la conexión a la base de datos desde todos los equipos se administrará un fichero de propiedades que el sistema leerá. De esta manera la aplicación permitirá o no realizar las diferentes funcionalidades dependiendo del rol de usuario que tenga el empleado que maneje la aplicación en ese momento.
- Esta aplicación proporcionará la información a la base de datos necesaria para el futuro etiquetado de los productos mediante el software independiente de la impresora.

Si se cumplen los objetivos marcados anteriormente descritos, se conseguirá:

- Mejorar el control del inventario de los productos del almacén.
- Introducir cantidades de productos para controlar la rotación del stock en la empresa.
- Conseguir localizar fácilmente los componentes en el almacén para su futura utilización.
- Gestión FIFO de los productos almacenados, para que no queden encasillados componentes antiguos en las estanterías del almacén.
- Normalización del sistema de etiquetado, creando uno propio para homogeneizar la identificación de los productos
- Cumplir con los estándares de calidad en el apartado de trazabilidad del producto comprendido en la norma ISO 9001:2015.

1.5. Requisitos del proyecto

Se van a distinguir dos tipos de requisitos para definir el trabajo: Requisitos de datos y requisitos funcionales.

- Los requisitos de datos son aquellos que van a abarcar el diseño y el desarrollo de la base de datos del SGA, así como posibles restricciones en los datos.
- Los requisitos funcionales serán los que definan la propia funcionalidad del proyecto. Es decir, qué pide el cliente (en este caso, la empresa) que haga la aplicación a diseñar y desarrollar.

1.5.1. Requisitos de datos

Se quiere diseñar una base de datos del almacén de componentes de la empresa Norpoo Prototipos S.L. Para comenzar a desarrollar el proyecto, la base de datos contará con los siguientes requisitos:

- Para cada producto que entra en el almacén se guardará un código con la referencia del producto, el albarán del pedido en el que ha venido, el nombre de su proveedor, una descripción indicando lo que es el componente, la fecha de fabricación del producto proporcionada por el proveedor (denominada DataCode y formada por un texto “MMM-YYYY”) y si está sujeto a condiciones de humedad.
- Hay productos que contienen la misma referencia, pero el albarán es diferente debido a los pedidos a proveedores que se realizan del mismo producto.
- Además queremos guardar, de cada producto, la ubicación desglosada en estantería, piso, fila y columna.
- También se desea almacenar la cantidad de unidades del producto, así como un apartado de comentarios, que pueden indicar la orden de fabricación donde se coloca este componente, si hay alguna incidencia en el producto, etc.
- Nos interesa guardar también aquellos productos que, en algún momento, pertenecieron al almacén. Es decir, que estuvieron en stock. Pero también nos interesa en qué fecha la cantidad del componente es 0. Por tanto queremos una tabla de productos fuera de stock, similar a la tabla de productos, cambiando el guardado de la cantidad (obviamente es 0), por la fecha en la que el producto entra en esa tabla. Servirá para mantener un histórico o log de productos que han estado en el almacén alguna vez.
- Existen productos que vienen en varios lotes, por tanto tienen como identificación la misma referencia y el mismo albarán. Es necesario que aquellos productos con esas características sean almacenados en un piso diferente, pero en la misma estantería.

1.5.2. Requisitos funcionales

Para el diseño y desarrollo de la aplicación principal del SGA que se desea construir en Norpoo Prototipos, se realizará lo siguiente:

- Tendrá como principales funcionalidades la búsqueda de producto por diferentes criterios, la inserción de productos, modificación de un producto y eliminación de este. Además existirá la función de muestra de un listado completo de toda la información de productos que se encuentran actualmente guardados, así como un desglose del stock disponible por referencia.
- La búsqueda de productos en la aplicación se divide en 3 apartados, según los criterios de la selección. Estos criterios son los siguientes:
    - Referencia del producto: Para la búsqueda de un producto por su referencia, se introducirá este identificador del producto. Tras introducirlo se mostrará en una tabla toda la información de ese componente. No será necesario introducir toda la referencia, debido a que suelen ser bastante largas y de esta manera se facilita el proceso de búsqueda.
    - Descripción del producto: Como en el apartado anterior, se introducirá parte de la descripción del producto por el mismo motivo que en la búsqueda por referencia. Después de introducirlo y ejecutar la acción, se mostrará todo producto que concuerde con toda, o parte de la descripción del producto proporcionada.
    - Identificación del proyecto: Cada proyecto electrónico que se realiza en la empresa requiere de una serie de componentes para ser realizado. Este proyecto recibe un identificador para separar los distintos productos de los diferentes proyectos. Su patrón es NPDXXXX. Siendo XXXX números del 0000 al 9999. Ej: NPD0123. Para encontrar aquellos productos pertenecientes a ese proyecto se introducirá el código completo NPDXXXX, y al ejecutar la acción se mostrará en la tabla correspondiente al resultado aquellos productos que estén contenidos en esa orden de trabajo.
    - Estos criterios han sido seleccionados de manera consensuada con la empresa, debido a que son la información que más se consulta durante el día en el área de producción.
- En cuanto a la inserción de un componente en la aplicación, se introducirá cada dato informativo en el campo que se pide. Como se ha comentado en una ocasión anterior, existen productos sensibles a las condiciones normales de temperatura y humedad de la empresa y se deben almacenar en una cámara especial. Estos productos se guardarán por defecto en la estantería AD101 si se ha insertado una S en el campo Humedad. En el caso de cometer un error escribiendo la S, pero colocando el producto en una estantería diferente a la AD101. Se parará la inserción al ejecutar la acción, notificando que los productos sujetos a condiciones especiales de atmósfera deben almacenarse en la estantería AD101, cambiando el campo de la estantería automáticamente a AD101 respectivamente, de esta manera, la inserción seguirá su funcionamiento normal.
- Para la modificación de los datos de los componentes, además de tener las consideraciones anteriores en cuenta, posee otras particularidades. Se debe introducir primeramente la referencia y el albarán del producto para identificarlo, así como seleccionar el campo a modificar e introducir el nuevo valor para el campo en cuestión.
- La eliminación sigue el mismo proceso que la modificación, pero con dos diferencias notables:

    - La primera, y más básica, es que no hay que seleccionar campos a modificar, debido a que es una eliminación del producto
    - La premisa más importante es que, al proceder con la eliminación del producto, se genera en la tabla del histórico de productos que han comprendido el almacén en algún momento una fila con los datos más importantes. Además de su ubicación para posibles retornos de componentes con la misma referencia. Es decir, no es una eliminación por completo del producto en el sistema, si no que se elimina de los productos activos en el almacén para ubicarse en el log de productos pasados.
- En cuanto a la muestra del listado, simplemente se ejecutará la acción y se mostrara todo el inventario activo en ese instante
- Para la acción de mostrar el stock de una referencia en concreto, se deberá introducir parte de esta, o la referencia entera, al igual que en el caso de buscar. Se mostrará en una tabla el stock disponible que hay en el almacén en ese momento.
- No todas las funciones estarán disponibles para todos los usuarios del programa. Se constituirá una jerarquía de roles, atendiendo a los puestos de trabajo que existen:
    - El rol más fuerte será el de Empleado de Almacén. Tendrá acceso y modificación de todas las tablas de la base de datos que lo conforma. Este usuario será el que tenga más permisos, debido a que es el encargado del almacén y todo el proceso y el desarrollo del programa tendrá su funcionamiento en el área de producción de la empresa
    - El siguiente rol será el de empleado, que podrá modificar la tabla de los productos, así como consultar las demás. Esto es debido a que, si se usa un producto, deberá de poder modificar tanto su cantidad, como su ubicación, incluso modificar algún aspecto de la información si existe algún error de tipografía en la descripción o en los comentarios.
    - El último rol será el de la Oficina, desde esta posición solo se podrá acceder a la información para consultas. No se realiza ninguna operación en la oficina de la empresa, por eso solo se tendrá la funcionalidad de búsquedas, listado y stocks.
- El etiquetado de los productos insertados para su ubicación en el almacén utilizará un software independiente del SGA, este software permite la conexión a la base de datos mediante SQL Server, y utilizará una colección de toda la información para imprimir etiquetas de manera dinámica. No se podrá editar la información desde ese software, solo acceder a ella. Por tanto, cualquier tipo de usuario podrá imprimir una etiqueta para un producto, ya sea de nueva entrada, modificado o un producto con etiqueta rota o desgastada.
- Se realizará semanalmente una copia de seguridad de toda la base de datos, cifrada y almacenada en Google Drive para su posible recuperación, para ello se usará el Software Duplicati 2.0.

De esta manera, se cumplirán todas las exigencias de la empresa para tener una trazabilidad y un seguimiento del almacén apropiado, además de satisfacer los estándares de la norma ISO 9001:2015.

1.6. Herramientas de trabajo

1.6.1. Sistema Gestor de Base de Datos

Se han valorado diferentes SGBD para su posible uso en el desarrollo de la base de datos de la aplicación. Son los siguientes:

- Microsoft SQL Server: Es un sistema gestor de bases de datos desarrollado por Microsoft. Permite la administración de servidores de datos y su lenguaje es Transact-SQL. Posee un entorno gráfico que ayuda a la definición de modelos, pero es limitado en cuanto a las plataformas donde se puede usar. Ofrece varios tipos de licencias. SQL Express, gratuita, nos proporciona lo necesario para el desarrollo del sistema.
- Oracle: Es una solución completa que incluye un motor de bases de datos, así como lenguaje para desarrollar tanto procedimientos almacenados como triggers. Posee una fuerte relación con Java y puede usarse en varias plataformas como Linux, Windows y Solaris. Su principal inconveniente es el altísimo precio de una licencia.
- MySQL: Es un SGBD conocido por su simplicidad y rendimiento. Hay características que no posee respecto a los otros sistemas mencionados, pero es una gran opción debido a que se trata de software libre y posee varias versiones dependiendo de las características de la base de datos.

El sistema elegido es SQL Server por las siguientes razones:

- Es un SGBD compatible, tanto con el lenguaje de programación a utilizar, como con el software independiente de la impresora.
- La velocidad de realización de operaciones es muy buena.
- Facilidad de instalación.
- La cantidad de datos a manejar en el caso de la empresa no es elevado, y con la versión gratuita, denominada SQL Express, nos es suficiente.

1.6.2. Entorno de desarrollo de la aplicación

Se ha elegido Java como lenguaje de desarrollo de la aplicación debido a su facilidad para utilizarlo en operaciones con bases de datos. Además de su múltiple funcionalidad, añadiendo también posibilidades de generar la interfaz gráfica del sistema e integrándolo con las operaciones anteriores.

Para la elección del entorno de desarrollo se han valorado las siguientes opciones:

- NetBeans: Desarrollado por Oracle, es un entorno de lo más completo y ampliable mediante módulos. Es un proyecto de código abierto y se puede instalar en cualquier sistema operativo en el que exista un JRE. Además es el primer entorno en el que se prueban las novedades del lenguaje Java.
- Eclipse: Es un proyecto de código abierto, como NetBeans. Está disponible para cualquier plataforma y posee una funcionalidad muy extensa, aunque tiene varios inconvenientes, como la complejidad de uso y el rendimiento pobre que puede dar en alguna instalación de sistema operativo.
- Intellij IDEA: Es un entorno desarrollado por JetBrains, soportando múltiples lenguajes, no es un proyecto totalmente basado en software libre, por lo que es de pago. Aunque está bien valorado por ser ágil y estable.

Se ha elegido NetBeans como entorno de desarrollo por los siguientes motivos:

- Facilidad de uso en lo referente al desarrollo de interfaces gráficas.
- Rapidez en la comunicación con SQL Server.
- IDE de código abierto.

1.7. Entregables

En un principio, los documentos y archivos a entregar serán los siguientes:

- Memoria del Proyecto: Contendrá todo el desarrollo del proyecto documentado.

- Anexos: En este entregable se incluirá la monitorización del proyecto, aspectos sobre el diseño, diagramas, etc.
- Aplicación: Conjunto de archivos que conforman la aplicación.

1.8. Diagrama EDT

En la ilustración 4 se muestra el diagrama de descomposición del trabajo del proyecto. En los Anexos se podrá observar con más detalle, debido a la extensión del gráfico.

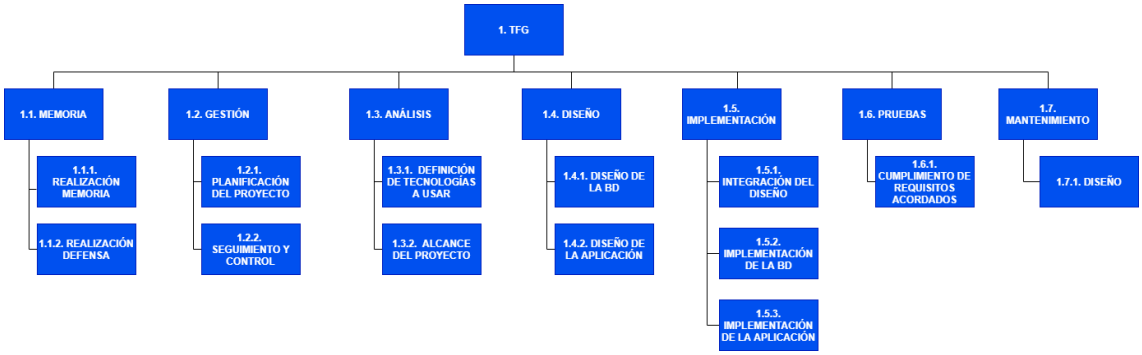

Ilustración 4. Diagrama EDT del proyecto.

TAREA		ENTREGABLE ASOCIADO
1.1.1.	Realización memoria	Memoria
1.1.2.	Realización defensa	Anexos
1.2.1.	Planificación del proyecto	Memoria
1.2.2.	Seguimiento y control	Anexos
1.3.1.	Definición de técnicas a usar	Memoria
1.3.2.	Alcance del proyecto	Memoria
1.4.1.	Diseño de la BD	Memoria
1.4.2.	Diseño de la aplicación	Memoria
1.5.1.	Integración del diseño	Aplicación
1.5.2.	Implementación de la BD	Aplicación
1.5.3.	Implementación de la aplicación	Aplicación
1.6.1.	Cumplimiento de requisitos	Memoria

1.7.1.	Diseño y elaboración manual de uso	Anexos

1.9. Planificación temporal

1.9.1. Descripción de actividades

Tarea 1.1.1. Realización memoria

En esta tarea se escribirá un documento con el desarrollo completo del proyecto, desde su etapa de planificación hasta su fase de mantenimiento.

Tarea 1.1.2. Realización defensa

Se preparará una presentación estimada de unos 10-12 minutos para su posterior defensa ante el tribunal de evaluación del proyecto.

Tarea 1.2.1. Planificación del proyecto

Dentro del documento de la memoria, se incluirá un apartado referente a la planificación detallada del proyecto, conteniendo un contexto inicial de la empresa y el alcance del trabajo a realizar. Además incluirá la gestión del tiempo, una evaluación de riesgos, criterios de calidad y un apartado valorando las partes interesadas del proyecto.

Tarea 1.2.2. Seguimiento y control

Se elaborará un documento durante toda la duración del proyecto que monitorice los tiempos marcados en la planificación anterior. El texto tendrá también consideraciones sobre las distintas desviaciones que se puedan dar: falta de tiempo, aparición de riesgos, etc.

Tarea 1.3.1. Definición de herramientas a usar

En el análisis del proyecto, dentro del alcance, se elabora una comparativa entre SGBDs y entornos de desarrollo. Se valoran 3 alternativas para cada una de las herramientas a utilizar y se elige una dependiendo de las necesidades y el rendimiento necesario para la aplicación propuesta.

Tarea 1.3.2. Alcance del proyecto

Esta tarea está incluida en la memoria del proyecto. Se basa en la definición del trabajo a realizar contemplando los requisitos del proyecto. Además contará con un contexto de la empresa, una situación inicial del problema y una serie de acciones previas para paliarlo. También posee un apartado de exclusiones para los requisitos que, por distintas razones, no se van a cumplir.

Tarea 1.4.1. Diseño de la BD

Dados los requisitos de datos en la tarea 1.3.2., se procederá a incluir en la memoria la fase completa del diseño de la base de datos para los productos de la empresa.

Tarea 1.4.2. Diseño de la aplicación

Dados los requisitos funcionales en la tarea 1.3.2., la memoria contendrá la fase completa del diseño de la aplicación de gestión del inventario.

Tarea 1.5.1. Integración del diseño

Completando las dos tareas anteriores, el diseño será aplicado para su implementación.

Tarea 1.5.2. Implementación de la BD

Con el diseño elaborado en la tarea 1.4.1., realizaremos la implementación de la base de datos en el SGBD elegido anteriormente.

Tarea 1.5.3. Implementación de la aplicación

Teniendo el diseño correctamente realizado en la tarea 1.4.2., comenzaremos a implementar la aplicación completa en el lenguaje y entorno de desarrollo elegido previamente.

Tarea 1.6.1. Cumplimiento de los requisitos

Al finalizar la fase de implementación del proyecto, se harán pruebas para comprobar que la aplicación funciona de la manera esperada. También verificaremos que los requisitos pedidos en un principio se cumplen.

Tarea 1.7.1. Elaboración manual de uso

Para evitar una mala utilización de la aplicación, crearemos un manual de uso de la aplicación.

1.9.2. Cronograma

Mediante un diagrama de Gantt se estimará en el tiempo el periodo de realización de las tareas descritas anteriormente. En los Anexos se observa el Diagrama de Gantt al completo y con más detalle.

		FEBRERO																								
TFG		4	5	6	7	8			11	12	13	14	15			18	19	20	21	22			25	26	27	28
1.1.1	Realización memoria																									
1.1.2	Realización defensa																									
1.2.1	Planificación del proyecto																									
1.2.2	Seguimiento y control																									
1.3.1	Definición de técnicas a usar																									
1.3.2	Alcance del proyecto																									
1.4.1	Diseño de la BD																									
1.4.2	Diseño de la aplicación																									
1.5.1	Integración del diseño																									
1.5.2	Implementación de la BD																									
1.5.3	Implementación de la aplicación																									
1.6.1	Cumplimiento de requisitos																									
1.7.1	Elaboración manual de uso																									

Ilustración 5. Diagrama de Gantt, mes de Febrero

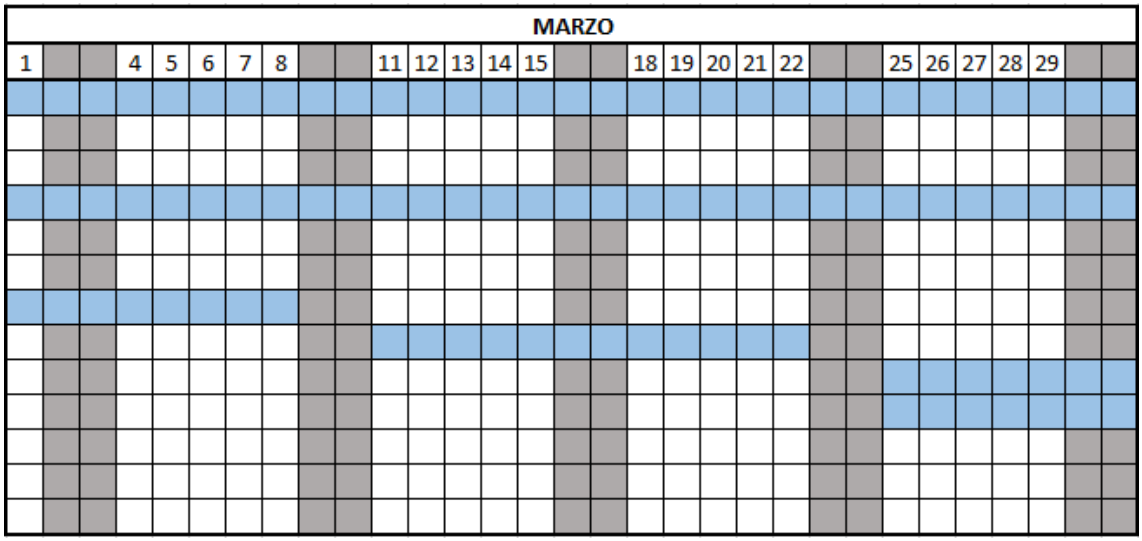

Ilustración 6. Diagrama de Gantt, mes de Marzo

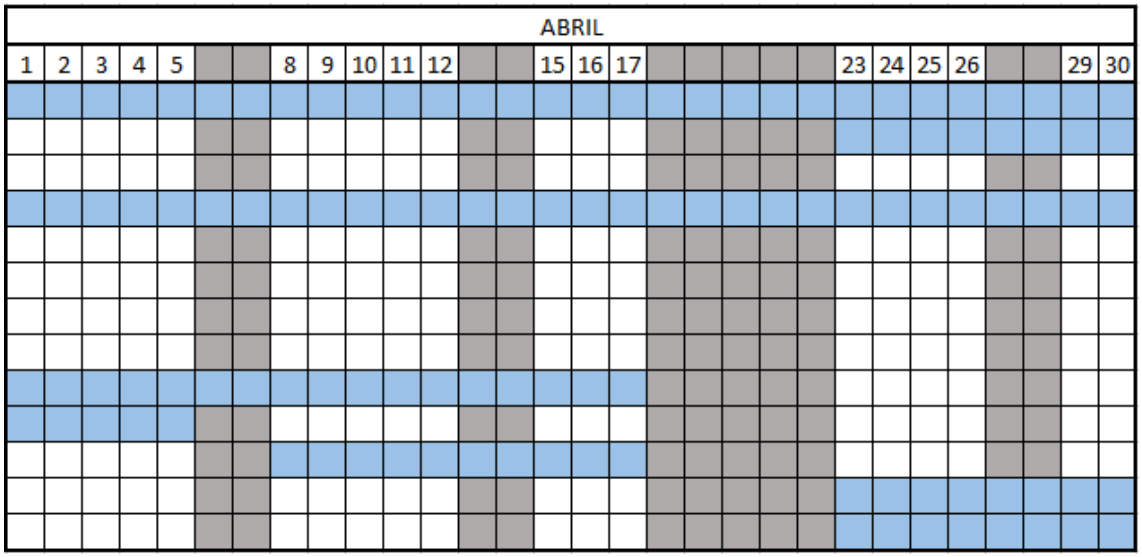

Ilustración 7. Diagrama de Gantt, mes de Abril

1.9.3. Dependencias

En la ilustración 8 se muestra el diagrama de dependencias del proyecto, se encuentra con más nitidez en los Anexos.

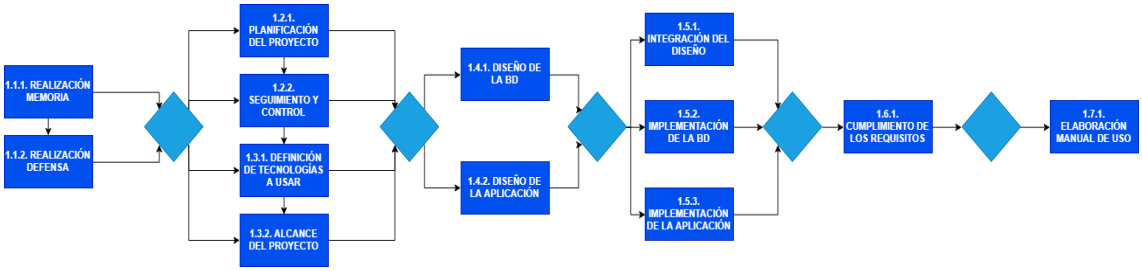

Ilustración 8. Diagrama de dependencias del proyecto

1.9.4. Hitos

Mediante un diagrama de hitos se estimará en el tiempo el cumplimiento de las tareas descritas anteriormente. En los Anexos se observa el Diagrama de Gantt al completo y con más detalle.

	FEBRERO																							
TFG	4	5	6	7	8			11	12	13	14	15			18	19	20	21	22			25	26	27	28
Comienzo realización memoria	◆																								
Inicio planificación del proyecto								◆																	
Inicio informe seguimiento y control								◆																	
Definición de técnicas a usar											◆														
Inicio alcance del proyecto												◆													
Fin definición alcance															◆										
Fin planificación del proyecto																			◆						
Inicio Diseño BD																						◆			
Fin Diseño BD																									
Inicio Diseño Aplicación																									
Fin Diseño Aplicación																									
Inicio integración diseño																									
Inicio Implementación BD																									
Fin Implementación BD																									
Inicio Implementación aplicación																									
Fin Implementación aplicación																									
Fin Integración Diseño																									
Inicio Cumplimiento de requisitos																									
Fin Cumplimiento de Requisitos																									
Inicio Elaboración manual de uso																									
Fin Elaboración manual de uso																									
Fin realización memoria																									
Comienzo realización defensa																									
Fin realización defensa																									

Ilustración 9. Diagrama de Hitos, mes de Febrero

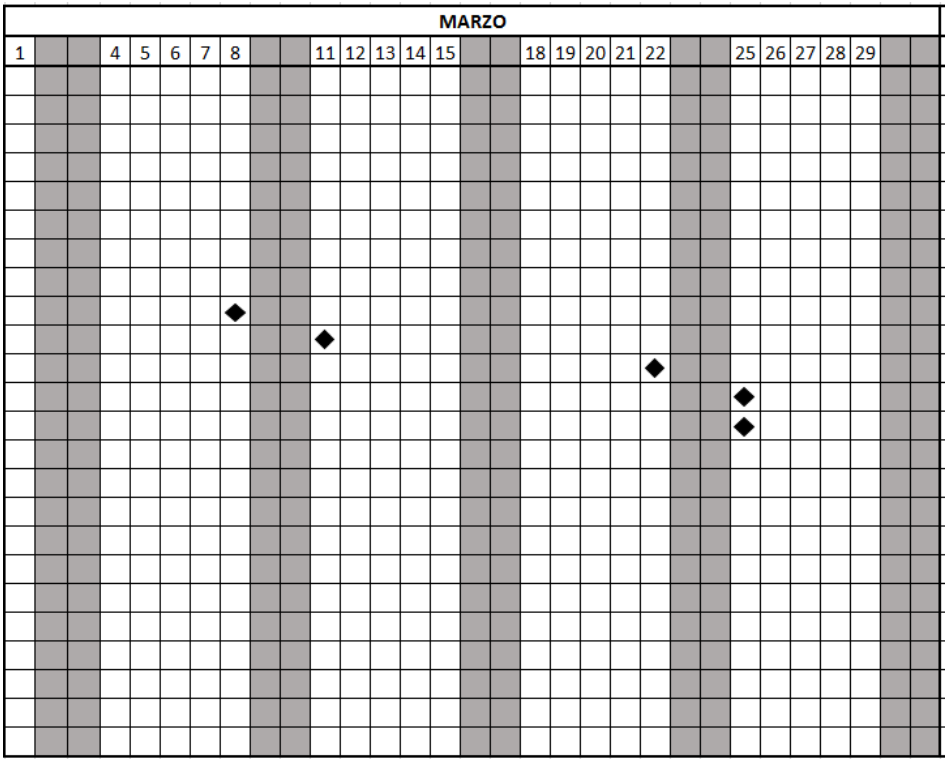

Ilustración 10. Diagrama de hitos, mes de Marzo

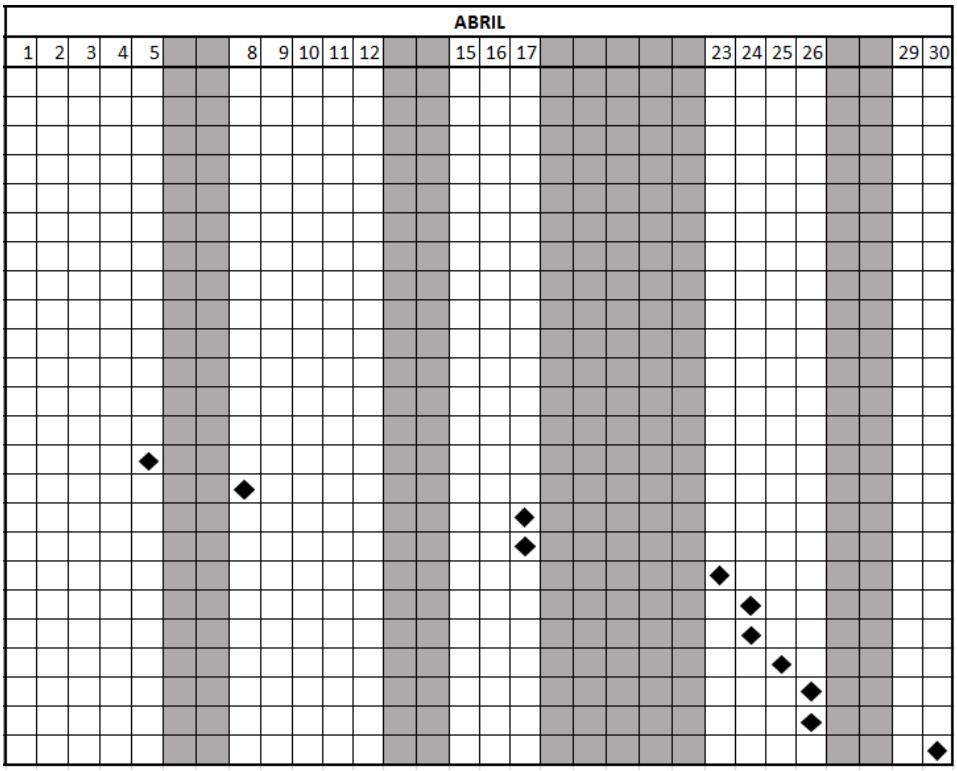

Ilustración 11. Diagrama de Hitos, mes de Abril

1.9.5. Estimaciones de dedicación

Tarea		Horas
1.1.1	Realización memoria	**50**
1.1.2	Realización defensa	**20**
1.2.1	Planificación del proyecto	**30**
1.2.2	Seguimiento y control	**15**
1.3.1	Definición de técnicas a usar	**5**
1.3.2	Alcance del proyecto	**10**
1.4.1	Diseño de la BD	**20**
1.4.2	Diseño de la aplicación	**30**
1.5.1	Integración del diseño	**20**

1.5.2	Implementación de la BD	**10**
1.5.3	Implementación de la aplicación	**60**
1.6.1	Cumplimiento de requisitos	**25**
1.7.1	Elaboración manual de uso	**5**
TOTAL HORAS:		**300**

1.10. Recursos materiales

Dispondremos de diferentes recursos materiales para la realización del proyecto, estos son los siguientes:

- Sistema Gestor de Bases de Datos SQL Server: Herramienta de creación, consulta y mantenimiento de bases de datos que utilizaremos para el almacenamiento de los datos del proyecto.
- Entorno de desarrollo NetBeans: Sistema con el que implementaremos y diseñaremos el código y las diferentes interfaces de la aplicación propuesta.
- Impresora Brother QL-700: Impresora de etiquetas con software para su diseño incorporado.

1.11. Comunicaciones

1.11.1. Medios de comunicación

Se utilizarán los siguientes medios de comunicación, tanto con el cliente (empresa) como con el tutor académico:

- Comunicación personal: Referente a distintas reuniones
- Correo electrónico: Para comunicarse tanto con el cliente como con el tutor académico

1.11.2. Comunicación interna

En un principio, se prevé que se usen los siguientes formatos para la documentación generada internamente:

- PDF o Word para documentación escrita.
- Archivos comprimidos en .zip o .rar para conjuntos de programas o documentos varios.

1.12. Riesgos

A raíz de los posibles cambios propuestos al principio del documento, se detectan varios riesgos. Clasificaremos los riesgos en 3 tipos para que sean evaluados.

- Riesgos referentes a Recursos Humanos: Aquellos que tienen que ver con la implicación personal de las partes interesadas en el proyecto.
- Riesgos referentes a las Relaciones con el Cliente: Aquellos que incumben a la comunicación con el cliente sobre aspectos del trabajo.
- Riesgos referentes a las Tecnologías: Aquellos que implican la posible disfunción por causas tecnológicas: fallos, desactualizaciones, etc.

Fuente	Riesgo	Si sucede...	Para evitarlo/minimizarlo...
Recursos Humanos	Disfunción en el rendimiento personal	Recortar el alcance	Seguir las pautas de la planificación e intentar ser más productivo
	Enfermedad/Imponderable	Reubicar tiempos de realización del trabajo	Intentar llevar el proyecto al día para no generar demasiada disfunción temporal
Relación con el Cliente	Disfunción en la comunicación con el cliente	Aprovechar al máximo las opciones de comunicación	Estar en continua comunicación con el cliente, si es posible
	Retrasos con el feedback del proyecto	Preguntar al cliente su opinión por el trabajo realizado	Cumplir con los plazos del proyecto para su correcto seguimiento.
	Cambios en los requisitos del cliente	Adecuar los requisitos al tamaño del proyecto	Evitar hacer más trabajo del previsto.
	Cambios en el alcance por parte del cliente	Adecuar el alcance conforme a los cambios exigidos	Realizar únicamente lo que se pida en el proyecto.
	Retrasos en el proyecto	Recortar el alcance	Seguir la planificación y la organización temporal propuesta en el proyecto.
Tecnología	No se puede usar alguna de las tecnologías previstas	Buscar otra alternativa si es fundamental para el desarrollo del proyecto, si no, se descarta	Estudio comparativo de tecnologías realizado
	Falta de tiempo para aprendizaje	Recortar el alcance del proyecto	Buscar información para tener nociones sobre la tecnología a usar
	Tecnología inadecuada para el proyecto	Buscar otra tecnología que se ajuste mejor al proyecto	Estudio comparativo de tecnologías realizado.

Además se pueden contemplar otros riesgos, como por ejemplo:

- No cumplir la legislación vigente debido su desconocimiento, en este caso se deberá estar al tanto de las leyes que puedan afectar al desarrollo del trabajo.
- Se pueden producir fallos en las tecnologías y el hardware utilizado, para ello se guardarán copias de seguridad del proyecto completo en diferentes sistemas.
- Puede haber errores en sistemas de comunicación utilizados debido a caídas de servidores. Ej.: Google

1.13. Partes interesadas

Para la definición del proyecto a llevar a cabo, las partes interesadas en que se realice son las siguientes:

- Cliente (empresa): Consideran el proyecto necesario para el aprovechamiento de los recursos materiales que se guardan en el almacén. Además es fundamental su funcionamiento para el cumplimiento de los estándares ISO ya conseguidos.
- Desarrollador del proyecto: Demostración, con este proyecto, que se cumplen las competencias aprendidas en los estudios realizados. Un buen resultado podría derivar en la incorporación en la empresa donde se desarrolla el proyecto.

2. Diseño

2.1. Diseño de la base de datos

2.1.1. Requisitos de datos

Los requisitos para el diseño de la base de datos han sido desglosados y explicados anteriormente en el apartado 2.5.1 de esta memoria.

2.1.2. Diagrama Entidad/Relación

En la Ilustración 12 se muestra el diagrama de entidad/relación extendido para el diseño de la base de datos que va a utilizar la aplicación. En los Anexos se colocará el diagrama para que se pueda observar con más nitidez, en caso de que sea necesario.

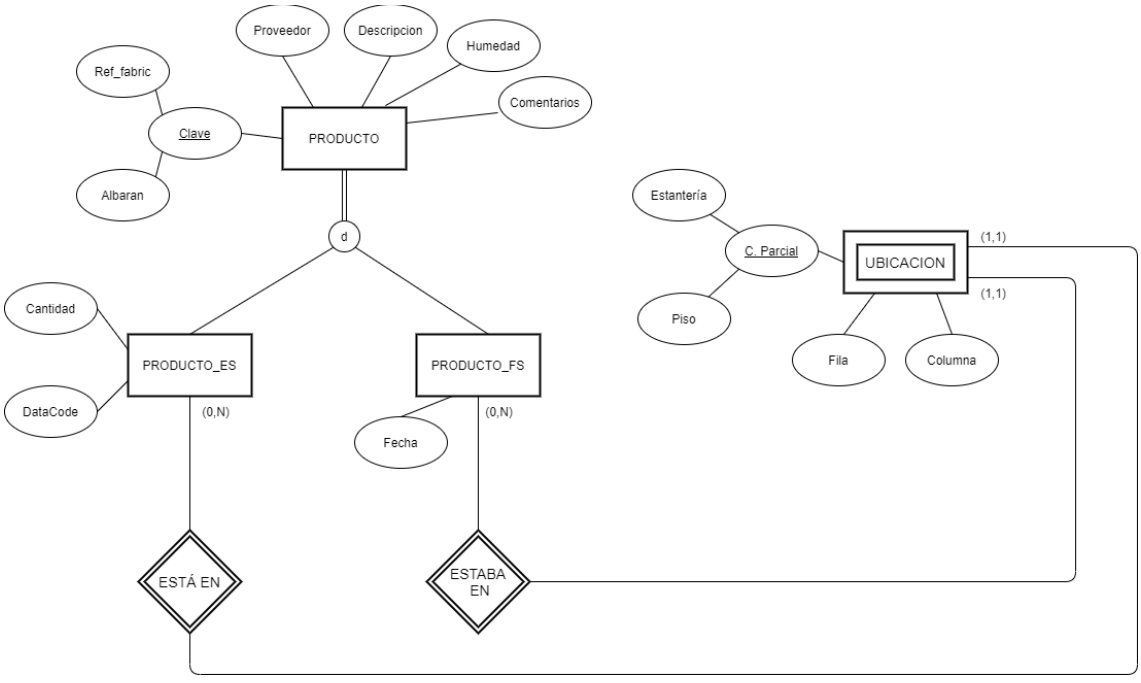

Ilustración 12. Diagrama Entidad/Relación de la base de datos

2.1.3. Diseño Lógico/Transformación a Modelo Relacional

Obteniendo el diagrama E/R, la transformación al modelo relacional para la base de datos del almacén de la empresa conformará de la siguiente manera:

PRODUCTO_ES

Ref_fabric	Albarán	Proveedor	DataCode	Descripción	Hum	Comentarios	Cantidad

PRODUCTO_FS

Ref_fabric	Albarán	Proveedor	Descripción	Hum	Comentarios	Fecha

UBICACION_PRODUCTO

Ref_fabric	Albarán	Estantería	Piso	Fila	Columna

CE: PRODUCTO_ES

UBICACION_FUERA_STOCK

Ref_fabric	Albarán	Estantería	Piso	Fila	Columna

CE: PRODUCTO_FS

2.1.4. Normalización

Todas las tablas de la base de datos de Norpoo Prototipos están normalizadas según la forma normal de Boyce-Codd. (Ver Anexos para consultar Normalización).

2.1.5. Diseño físico

Las tablas de la base de datos que se ha generado tienen pocas filas de datos (800 aproximadamente a día 4 de Mayo de 2020). Es por esto que no es necesario realizar un diseño físico debido a que no hacen falta índices secundarios para acceder a los datos de manera más rápida.

2.2. Diseño de la Aplicación

2.2.1. Diagrama de casos de uso

En la ilustración 13 se muestra el diagrama de casos de uso generado para la aplicación después de analizar los requisitos funcionales.

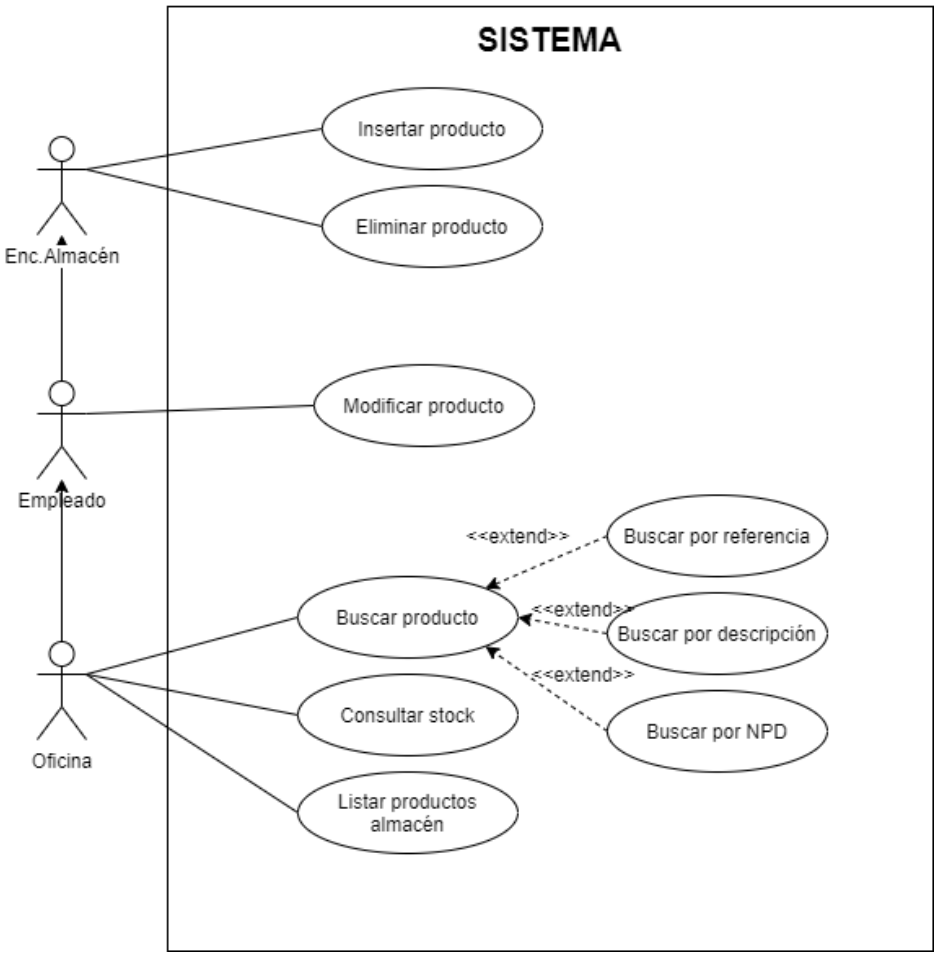

Ilustración 13. Diagrama de casos de uso de la aplicación

2.2.2. Especificación de los casos de uso

Con el diagrama de casos de uso visto en la ilustración anterior, procederemos a especificar cada caso de uso para entenderlo correctamente. Usaremos una plantilla Larman para realizar este apartado.

No se incluirán todas las especificaciones de los casos de uso en la memoria, se incluirán varios ejemplos. Pero la totalidad se encontrará en los anexos.

Caso de uso: Insertar producto
Objetivo: Añadir un producto adquirido recientemente a la base de datos del almacén de la empresa para su colocación en las estanterías
Actores: Empleado Almacén (A)
Precondiciones: -
Pasos: 1- A: Inicio de la aplicación 2- A: Seleccionar opción de insertar producto 3- S: Se muestra pantalla de inserción del producto 4- A: Insertar campos del producto 5- A: Seleccionar botón de insertar 6- S: Mensaje de confirmación de inserción 7- S: Mensaje de producto insertado
Extensiones: 6.1 - Selección de aceptar en mensaje de confirmación. 6.1.1 - Mensaje de error en la inserción de producto 6.1.1.1 - Regresar al paso 3. 6.1.2. - Mensaje de error en la conexión a la base de datos. 6.1.2.1 - Regresar al paso 3.
Cuestiones:

Ilustración 14. Caso de Uso Insertar Producto

Caso de uso: Modificar producto
Objetivo: Modificar algún campo de un producto que se encuentra en la base de datos del almacén de la empresa
Actores: Empleado (E)
Precondiciones: -
Pasos: 1- E: Inicio de la aplicación 2- E: Seleccionar opción de modificar producto 3- E: Seleccionar campo a modificar 4- S: Se muestra pantalla de modificación del campo de producto seleccionado 5- E: Insertar campos identificativos del producto a modificar 6- E: Insertar campo modificado 7- E: Seleccionar botón de modificar 8- S: Mensaje de confirmación de modificación 9- S: Mensaje de producto modificado
Extensiones: 8.1 - Selección de aceptar en el mensaje de confirmación 8.1.1 - Error en la identificación del producto (Campos incorrectos, producto no existente…) 8.1.1.1 - Regresar al paso 4 8.1.2. - Error en la modificación del producto 8.1.2.1 - Regresar al paso 4 8.1.3. - Mensaje de error de conexión a la base de datos 8.1.3.1 - Regresar al paso 4.
Cuestiones:

Ilustración 15. Caso de Uso Modificar Producto

Caso de uso: Buscar producto
Objetivo: Encontrar un producto existente en la base de datos de la aplicación del almacén de la empresa
Actores: Usuario Oficina (O)
Precondiciones: -
Pasos: 1- O: Inicio de la aplicación 2- O: Seleccionar opción de Búsqueda de producto 3- O: Seleccionar opción de tipo de búsqueda 4- S: Buscar producto
Extensiones: 3.1. - Selección de opción Listado de productos 3.1.1 - Caso de Uso Listado de Productos 3.2. - Selección de opción Búsqueda por Referencia 3.2.1. - Caso de Uso Búsqueda por referencia 3.3 – Selección de opción Búsqueda por descripción 3.3.1 - Caso de Uso Búsqueda por Descripción
Cuestiones:

Ilustración 16. Caso de Uso Buscar Producto

Caso de uso: Búsqueda por referencia
Objetivo: Mostrar el producto o productos con una referencia en concreto.
Actores: Usuario oficina (O)
Precondiciones: -
Pasos: 1- S: Caso de Uso Búsqueda de Producto 2- U: Introducir referencia del fabricante del producto 3- S: Muestra del producto o productos con la referencia introducida
Extensiones: 3.1. - Error de introducción de campos (Referencia no válida) 3.1.1. - Mensaje de error indicando que la referencia no es válida 3.1.2. - Regresar al paso 2 3.2 - No existe ningún producto con la referencia introducida 3.2.1 - Mensaje indicando que no existe un producto con la referencia introducida 3.2.2 - Regresar al paso 2

Ilustración 17. Caso de uso Búsqueda por Referencia

2.2.3. Diagrama de Clases de la aplicación

En la ilustración 18 se mostrará el diagrama de clases de la aplicación a implementar. En los anexos se podrá observar con más nitidez.

Se observa que la aplicación se va a implementar por capas. Este diseño se explicará en el apartado de implementación.

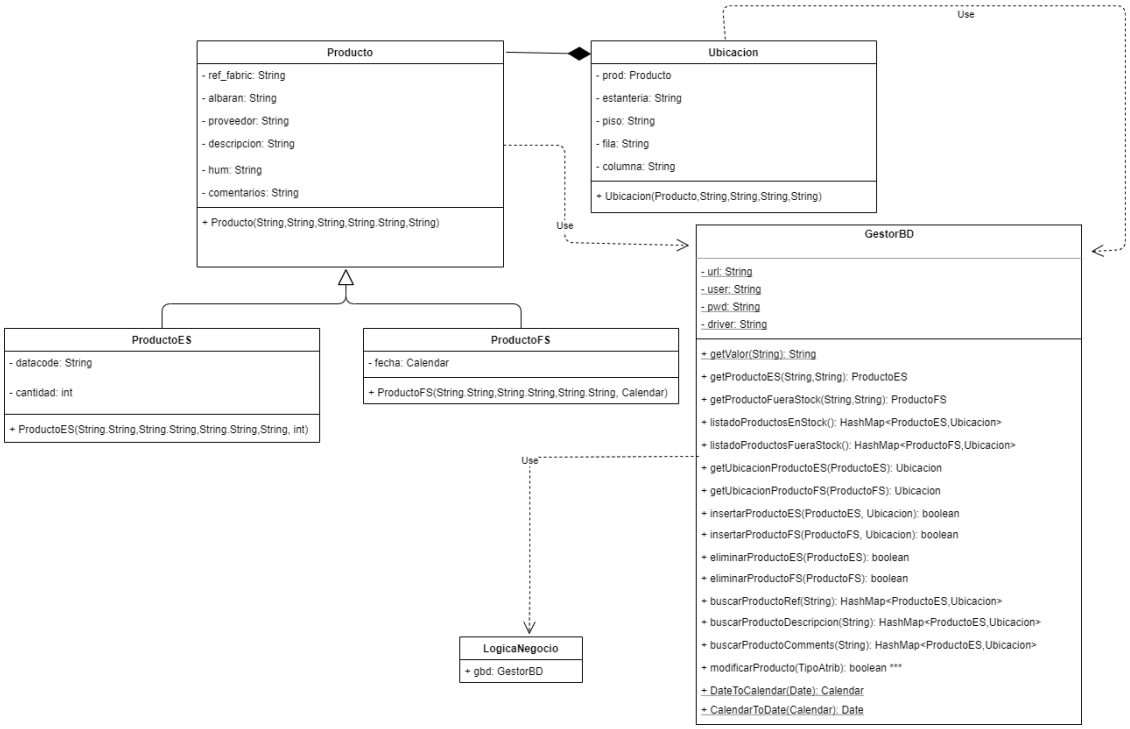

Ilustración 18. Diagrama de clases de la aplicación

2.2.4. Diagramas de actividad

A continuación se mostrará, mediante varios diagramas de actividad, la funcionalidad de algunas opciones del sistema.

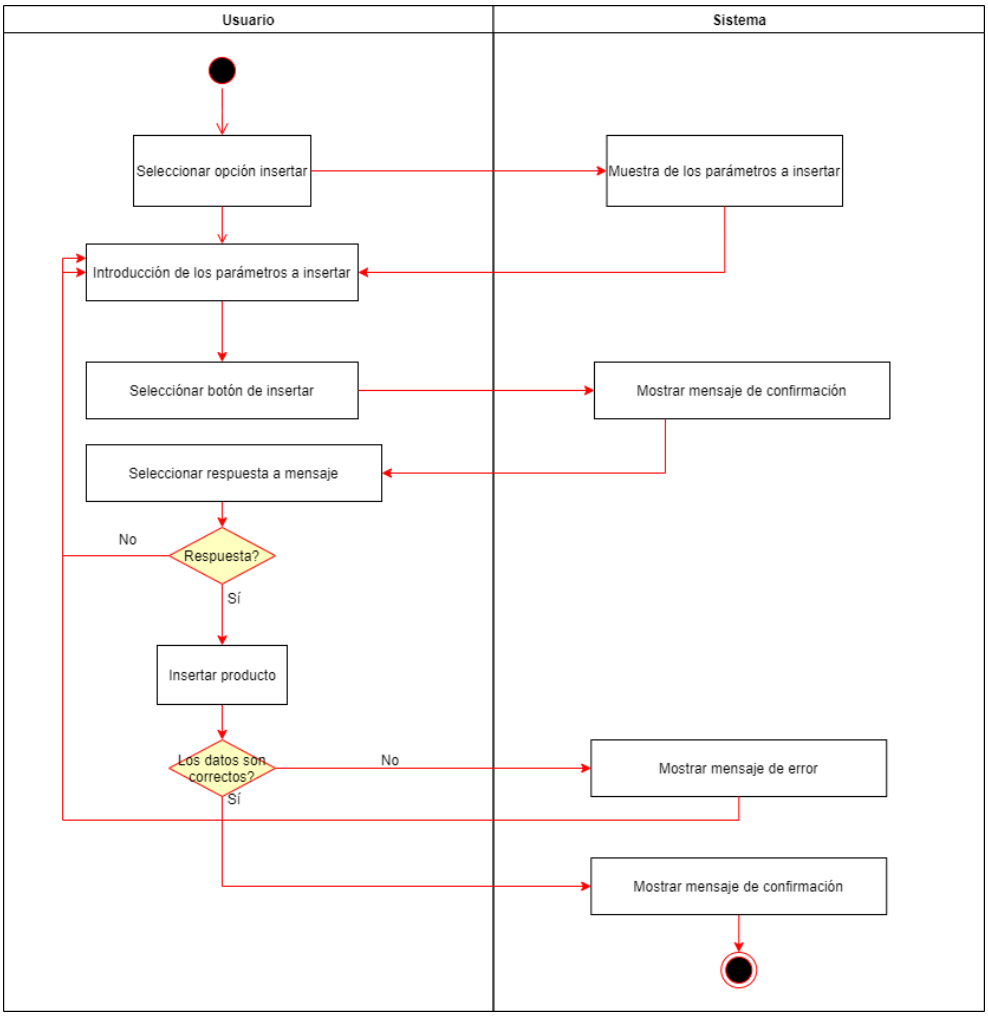

Ilustración 19. Diagrama de actividad Insertar producto

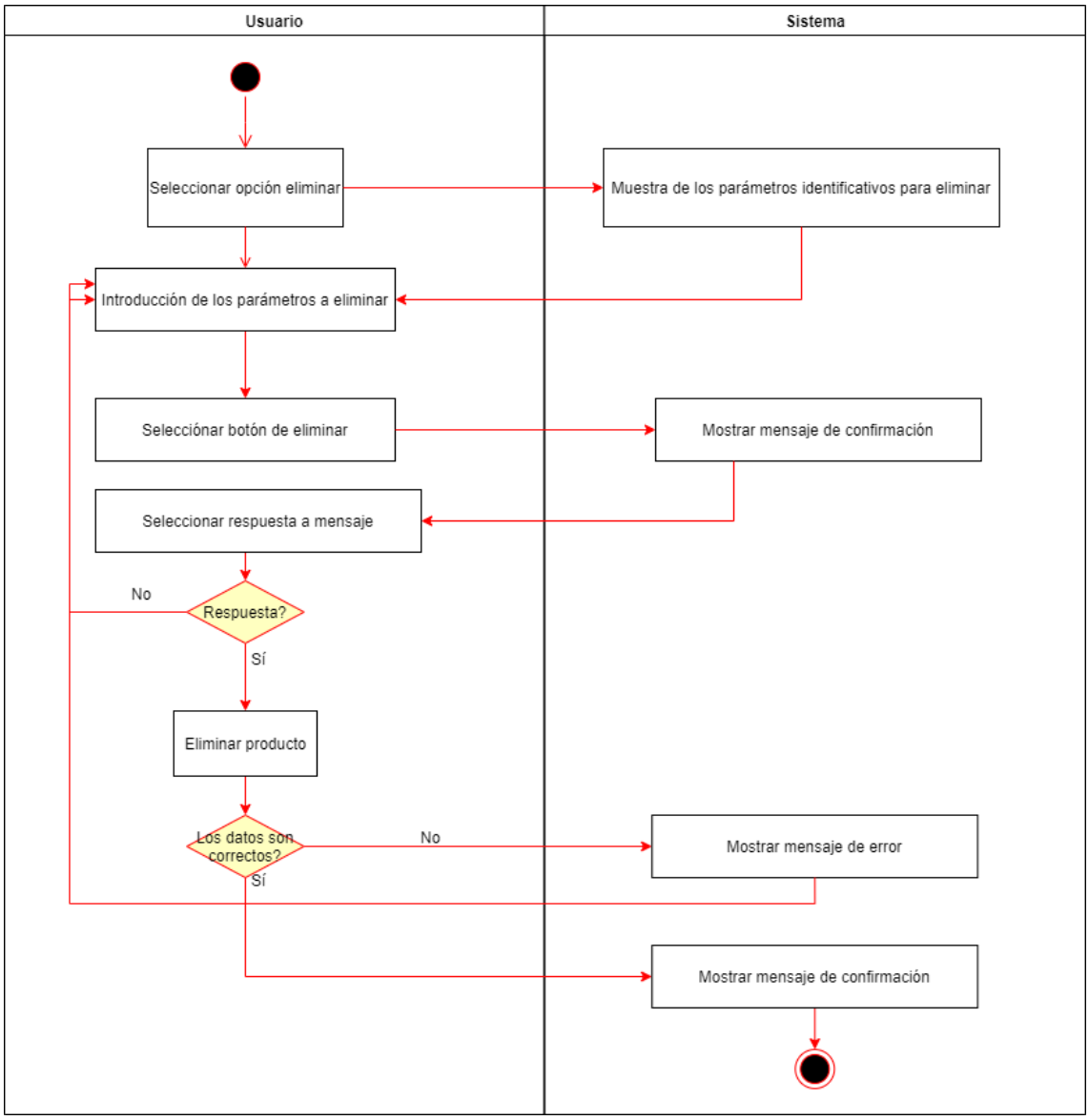

Ilustración 20. Diagrama de actividad eliminar Producto

2.2.5. Diseño de la interfaz gráfica

Para el diseño de la interfaz gráfica se ha utilizado NetBeans. Este IDE, además de ser el utilizado para la implementación del SGA, nos da la posibilidad de añadir interfaces gráficas a nuestros proyectos.

A continuación se mostrará una imagen de 2 de las opciones que contiene la aplicación. El resto de interfaces son similares y se encontrarán en los anexos.

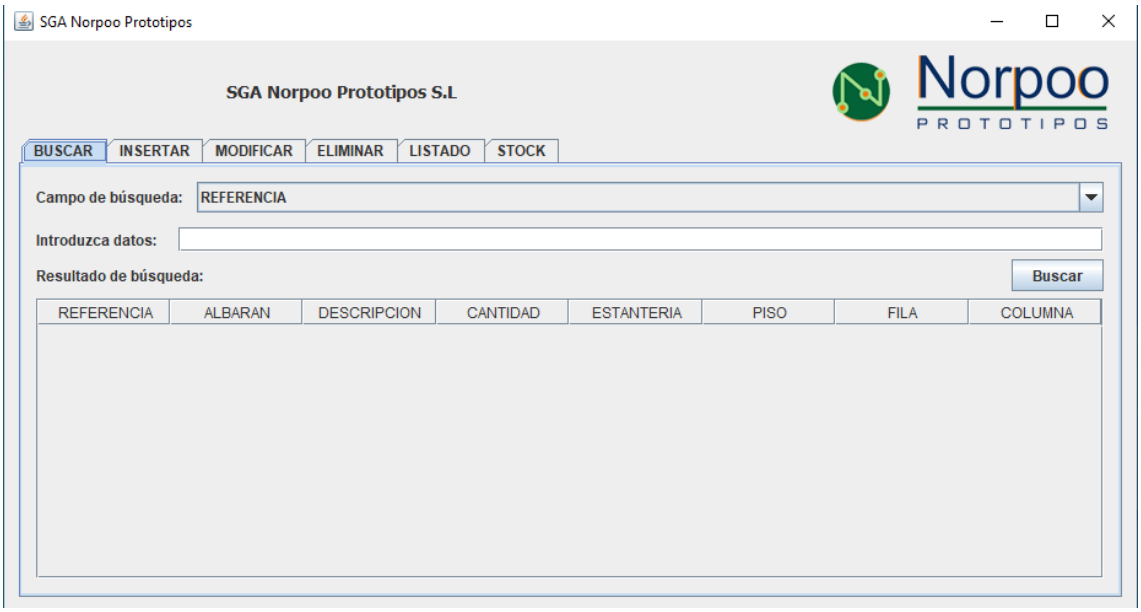

Ilustración 21. Interfaz gráfica búsqueda por referencia

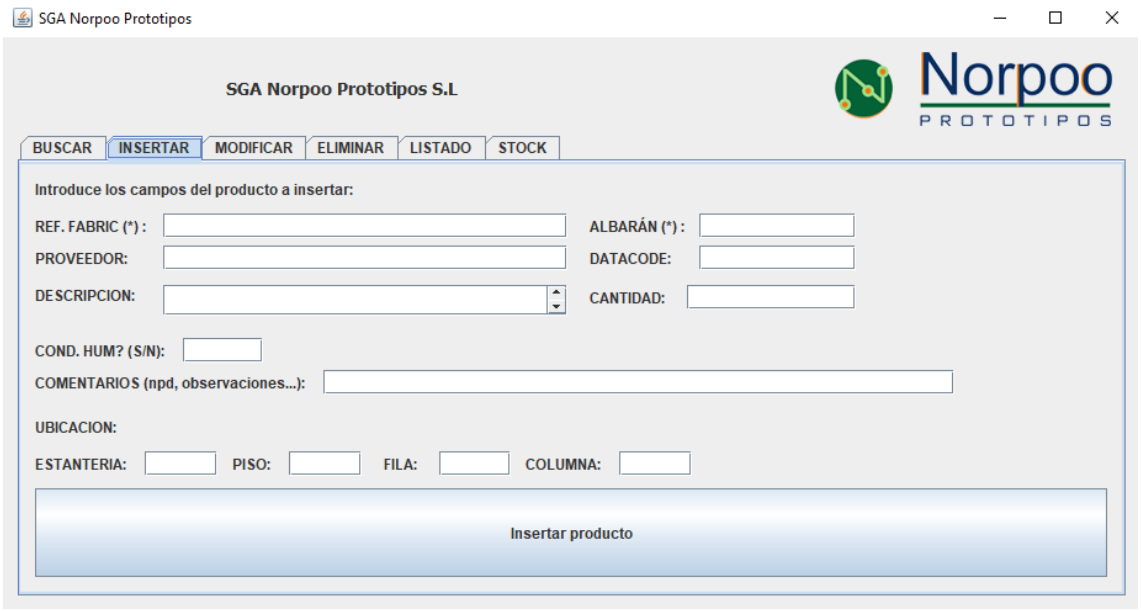

Ilustración 22. Interfaz gráfica Insertar producto

3. Implementación

3.1. Implementación de la base de datos

Como hemos comentado anteriormente en el análisis de tecnologías a utilizar, se utilizará el SGBD SQL Server. Para empezar la implementación utilizaremos el software SQL Server Management Studio. Nos centraremos en 3 apartados fundamentales en la implementación. La creación de la base de datos, que se ocupará de mostrar cómo se crean las tablas y sus restricciones. La creación de usuarios, que se va a encargar de definir los roles de los manipuladores de los datos de las tablas y la introducción de datos.

3.1.1. Creación de la base de datos

Para crear la base de datos, debemos utilizar el usuario raíz del SGBD, de esta manera, podremos asignar espacio a los archivos para las tablas y para el historial de cambios.

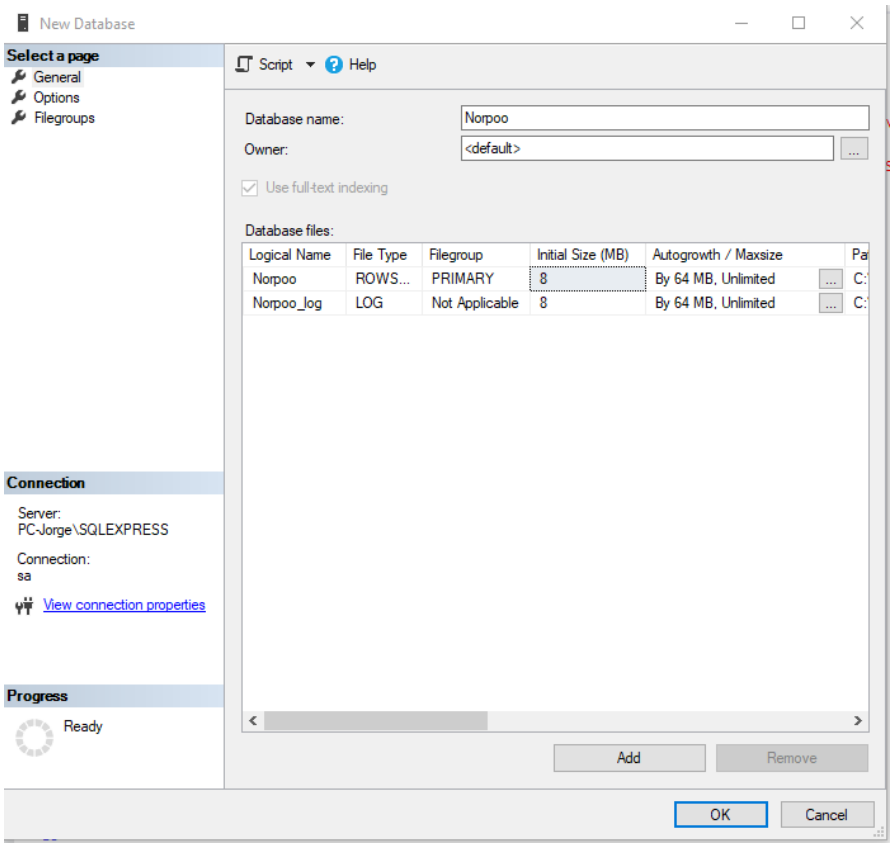

Ilustración 23. Interfaz de creación de la Base de Datos

Tras generar la base de datos vacía, se procederá a la creación de las tablas que van a componer nuestro almacén.

3.1.1.1. Creación de las tablas

Crearemos las tablas siguiendo el modelo entidad-relación desarrollado anteriormente en el punto 3.1.2. . La base de datos se compone de 4 tablas relacionadas entre sí, donde los campos tienen un tipo determinado para la correcta manipulación de estos cuando haya datos.

A continuación mostraremos el código de creación de las tablas, y en el siguiente punto trataremos el tema de la adición de restricciones.

```
CREATE TABLE [norpoo].[producto_es](
    [REF_FABRIC] [nchar](30) NOT NULL,
    [ALBARAN] [nchar](30) NOT NULL,
    [PROVEEDOR] [nchar](30) NULL,
    [DATACODE] [nchar](30) NULL,
    [DESCRIPCION] [nchar](250) NULL,
    [HUM] [nchar](1) NULL,
    [COMENTARIOS] [nchar](250) NULL,
    [CANTIDAD] [int] NULL
) ON [PRIMARY]
GO
```

Ilustración 24. Creación de la tabla PRODUCTO_ES

```
CREATE TABLE [norpoo].[ubicacion_producto](
    [REF_FABRIC] [nchar](30) NOT NULL,
    [ALBARAN] [nchar](30) NOT NULL,
    [ESTANTERIA] [nchar](10) NOT NULL,
    [PISO] [int] NOT NULL,
    [FILA] [int] NULL,
    [COLUMNA] [int] NULL
) ON [PRIMARY]
GO
```

Ilustración 25. Creación de la tabla UBICACION_PRODUCTO

```
CREATE TABLE [norpoo].[producto_fs](
    [REF_FABRIC] [nchar](30) NOT NULL,
    [ALBARAN] [nchar](30) NOT NULL,
    [PROVEEDOR] [nchar](30) NULL,
    [DESCRIPCION] [nchar](250) NULL,
    [HUM] [nchar](1) NULL,
    [COMENTARIOS] [nchar](250) NULL,
    [FECHA] [date] NULL
) ON [PRIMARY]
GO
```

Ilustración 26. Creación de la tabla PRODUCTO_FS

```
CREATE TABLE [norpoo].[ubicacion_fuera_stock](
    [REF_FABRIC] [nchar](30) NOT NULL,
    [ALBARAN] [nchar](30) NOT NULL,
    [ESTANTERIA] [nchar](10) NOT NULL,
    [PISO] [int] NOT NULL,
    [FILA] [int] NULL,
    [COLUMNA] [int] NULL
) ON [PRIMARY]
GO
```

Ilustración 27. Creación de la tabla UBICACIÓN_FUERA_STOCK

3.1.1.2. Restricciones de tablas

Las cuatro tablas descritas en el apartado anterior están creadas primeramente sin restricciones. Es por eso que, para preservar la integridad de los datos y para que en un futuro no haya incongruencias en la información, debemos crear restricciones de clave, tanto para las claves primarias como para las claves foráneas.

En SQL Server nos surge la dificultad de que, si asignamos una clave primaria múltiple, debemos hacerlo mediante un índice non-clustered, debido a que el índice clustered que crea las claves por defecto tiene un tamaño máximo de contenido de datos.

Las restricciones de clave primaria para las 4 tablas son de la manera mostrada en las siguientes ilustraciones.

```
ALTER TABLE [norpoo].[producto_es] ADD  CONSTRAINT [PK_producto_es_REF_FABRIC] PRIMARY KEY NONCLUSTERED
(
    [REF_FABRIC] ASC,
    [ALBARAN] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, ONLINE = OFF, ALLO
GO
SET ANSI_PADDING ON
GO
```

Ilustración 28. Restricción de clave primaria en tabla PRODUCTO_ES

```
ALTER TABLE [norpoo].[producto_fs] ADD  CONSTRAINT [PK_producto_fs_REF_FABRIC] PRIMARY KEY NONCLUSTERED
(
    [REF_FABRIC] ASC,
    [ALBARAN] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, ONLINE = OFF, ,
GO
SET ANSI_PADDING ON
GO
```

Ilustración 29. Restricción de clave primaria en tabla PRODUCTO_FS

```
ALTER TABLE [norpoo].[ubicacion_fuera_stock] ADD  CONSTRAINT [PK_ubicacion_fuera_stock_REF_FABRIC] PRIMARY KEY NONCLUSTERED
(
    [REF_FABRIC] ASC,
    [ALBARAN] ASC,
    [ESTANTERIA] ASC,
    [PISO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS
GO
```

Ilustración 30. Restricción de clave primaria en la tabla UBICACIÓN_FUERA_STOCK

```
ALTER TABLE [norpoo].[ubicacion_producto] ADD  CONSTRAINT [PK_ubicacion_producto_REF_FABRIC] PRIMARY KEY NONCLUSTERED
(
    [REF_FABRIC] ASC,
    [ALBARAN] ASC,
    [ESTANTERIA] ASC,
    [PISO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, ONLINE = OFF, ALLOW_ROW
GO
```

Ilustración 31. Restricción de clave primaria en la tabla UBICACIÓN_PRODUCTO

Estas restricciones garantizan que los datos no se repitan en las tablas. Pero, aunque parezcan que las claves sean similares, estas tablas no están relacionadas todavía, y no hay una conexión entre tablas para hacer que el sistema funcione bajo los estándares del modelo relacional. Para ello, según el modelo relacional desarrollado, existen dos tablas con clave foránea hacia las otras 2. Debemos generar esas restricciones para que no se pierda información en actualizaciones o se repita información en inserciones.

Las restricciones en este caso son simples, debemos referenciar, en la tabla correspondiente, los campos REF_FABRIC y ALBARAN, que son los campos principales de la base de datos que vamos a utilizar para manejar el almacén de la empresa.

Es decir, uniremos en una restricción los campos mencionados entre las tablas PRODUCTO_ES y UBICACION_PRODUCTO. Además de en PRODUCTO_FS y UBICACION_FUERA_STOCK. Queremos que la actualización de los datos en caso de modificaciones, inserciones o eliminaciones sea en cascada, es decir, que un cambio en los datos genera automáticamente los cambios en las tablas relacionadas. Por tanto, las restricciones de clave foránea quedan de la siguiente manera:

```
ALTER TABLE [norpoo].[ubicacion_fuera_stock]  WITH NOCHECK ADD  CONSTRAINT [ubicacion_fuera_stock$ubicacion_fuera_stock_ibfk_1] FOREIGN KEY([REF_FABRIC], [ALBARAN])
REFERENCES [norpoo].[producto_fs] ([REF_FABRIC], [ALBARAN])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [norpoo].[ubicacion_fuera_stock] NOCHECK CONSTRAINT [ubicacion_fuera_stock$ubicacion_fuera_stock_ibfk_1]
GO
ALTER TABLE [norpoo].[ubicacion_producto]  WITH NOCHECK ADD  CONSTRAINT [ubicacion_producto$ubicacion_producto_ibfk_1] FOREIGN KEY([REF_FABRIC], [ALBARAN])
REFERENCES [norpoo].[producto_es] ([REF_FABRIC], [ALBARAN])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
```

Ilustración 32. Restricciones de clave foránea en las tablas UBICACION_PRODUCTO y UBICACION_FUERA_STOCK

3.1.1.3. Creación de usuarios

Para respetar los roles en la empresa comentados en el análisis, debemos crear una serie de usuarios, de esta manera, protegemos la base de datos ante cambios indeseados y cada usuario tendrá los privilegios suficientes para realizar su cometido en la empresa correctamente.

Debemos crear usuarios que utilicen la base de datos de la empresa, debemos crear un login, dar una contraseña para ese login y administrar los permisos para ese inicio de sesión, que será el utilizado correspondientemente en cada equipo para acceder a la funcionalidad del sistema apropiada.

En la ilustración 33 mostraremos un ejemplo de rol de usuario y su configuración. En este caso, el usuario oficina.

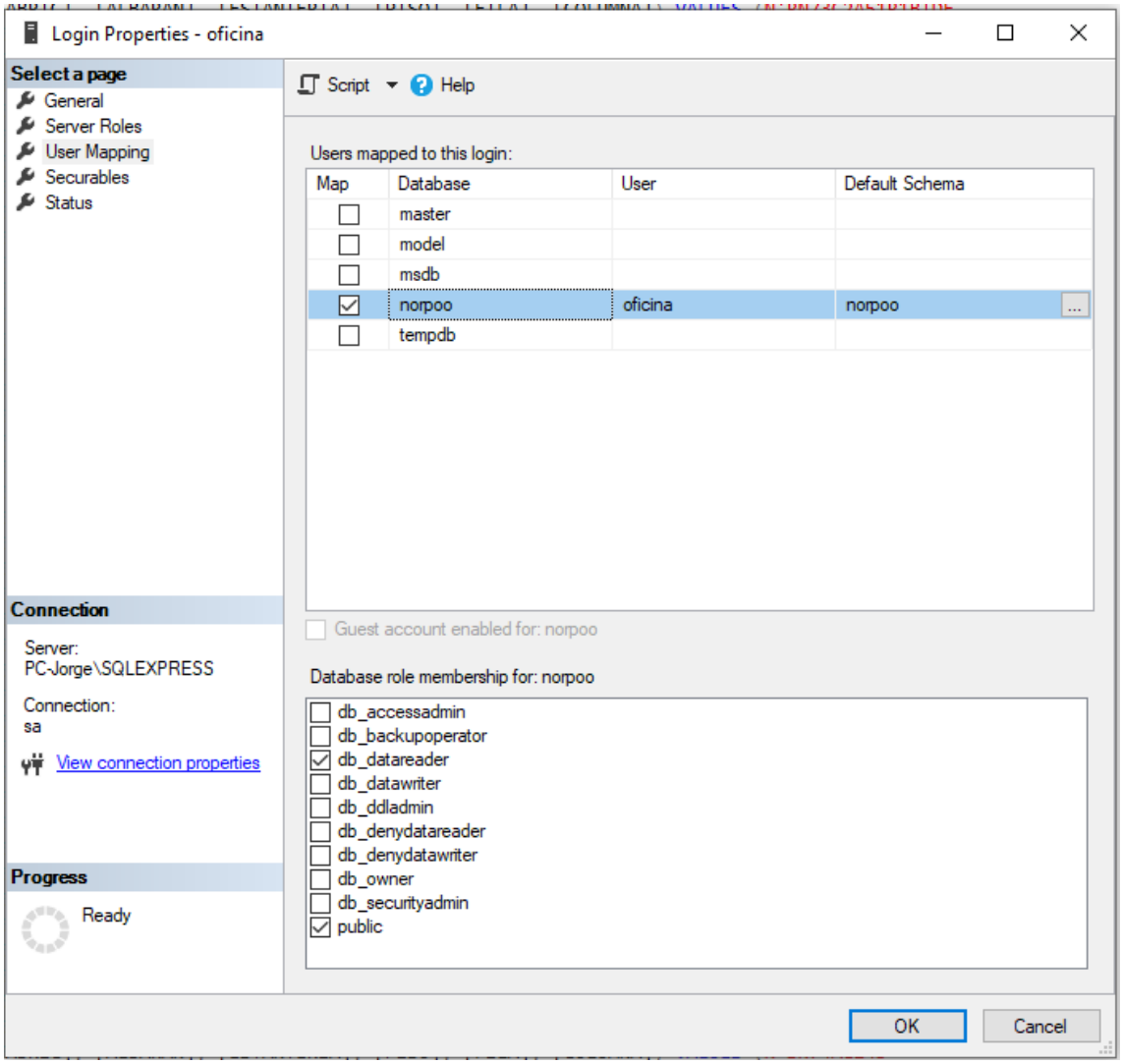

Ilustración 33. Selección de permisos del usuario oficina para la base de datos Norpoo

3.1.1.4. Inserción de los datos

Para la introducción de los datos, en este caso, los hemos introducido a mano. La base de datos en su primer momento tenía unas 150 filas. Por tanto, no fue necesario crear un script de introducción de la información.

A partir de ese momento, la manipulación de los datos se realizará mediante la aplicación de escritorio desarrollada en este proyecto.

3.2. Implementación de la aplicación

Después de implementar la base de datos y generar la vista para las etiquetas, procederemos al desarrollo del grueso de la aplicación. Para ello se utilizará NetBeans como se ha comentado anteriormente.

El proceso que se va a seguir para la implementación es el modelo por capas. Se desarrollarán de manera gradual 4 capas, estas son:

- Modelo de datos: Creación de los objetos a manipular por la base de datos
- Capa de persistencia: Usa el modelo de datos para realizar las operaciones CRUD, que son las operaciones básicas a realizar en una base de datos.
- Lógica de Negocio: Conecta la capa de persistencia junto con la Interfaz gráfica proporcionándole los servicios necesarios para que esta funcione correctamente.
- Capa de Presentación: Conjunto de interfaces gráficas que el usuario podrá manejar para obtener toda la funcionalidad del sistema.

Explicaremos el desarrollo de cada capa de manera detallada, indicando las posibles dificultades que han ido surgiendo a lo largo de la implementación.

3.2.1. Implementación del modelo de datos

Para la implementación del modelo de datos nos basaremos en el diagrama de clases de la aplicación. Según este diagrama y la base de datos nos surgen 4 objetos principales a construir para manipular correctamente el SGA. Estos objetos principales son:

- Producto: Clase troncal que abarca todos los componentes que han estado o están actualmente en el almacén de la empresa. Esta clase recoge todos los campos comunes entre los dos tipos de objetos que se pueden dar. Estos objetos los explicaremos a continuación.
- Producto_ES: Producto que está actualmente en las estanterías del almacén de la empresa.
- Producto_FS: Producto que se encuentra dado de baja en la aplicación
- Ubicación: Posición que ocupa o ha ocupado un componente en las estanterías.

3.2.1.1. Clase Producto

La clase producto es la clase troncal de la aplicación. Es la que modela los objetos que se encuentran en el almacén de la empresa. Sobre ella se modelan los dos tipos de producto que existen en el sistema. En esta clase se recogen los atributos que tienen en común ambos tipos de objetos.

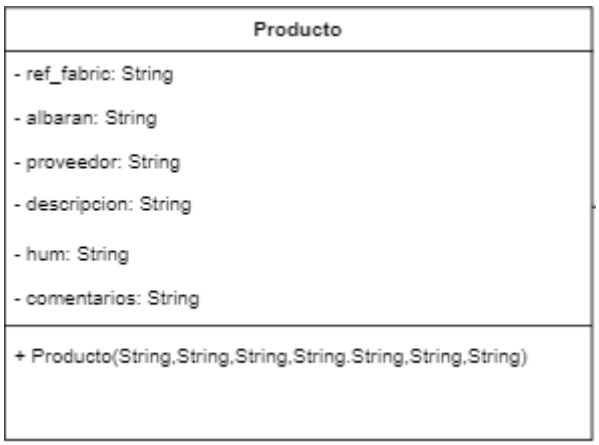

Ilustración 34. Clase Producto

En la ilustración 34 se observa la clase producto con los atributos comunes a los dos tipos que ella contiene, así como su constructor. No se permite un constructor sin parámetros, debido a que todo producto debe tener al menos una referencia y un albarán.

Cada uno de los atributos de la clase tiene un getter y un setter, que se estructuran de la misma manera en todas las clases del modelo. Es decir, en las clases restantes de explicar también funcionan de la misma manera:

```
/**
 * Método que devuelve la referencia del fabricante del producto solicitado.
 *
 * @return Referencia del fabricante del producto
 *
 */
public String getRef_fabric() {
    return this.ref_fabric;
}

/**
 * Método que modifica la referencia del fabricante del producto solicitado.
 *
 * @param ref El parámetro ref define la referencia del fabricante
 * perteneciente al producto
 *
 */
public void setRef_fabric(String ref) {
    this.ref_fabric = ref;
}
```

Ilustración 35. Getter y setter de la clase Producto

3.2.1.2. Clase ProductoES

La clase ProductoES modela aquellos componentes que actualmente se encuentran en el almacén de la empresa. Se diferencia de la clase superior en 2 matices, es por esto que se ha creado la división de las clases.

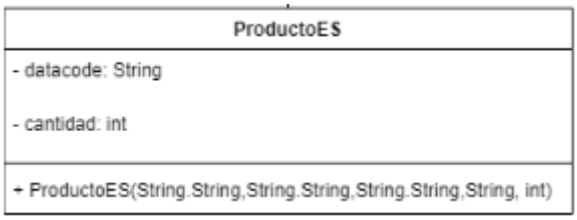

Ilustración 36. Clase ProductoES

El primer detalle es que se desea guardar el datacode del producto en el almacén. Este datacode indica la fecha en el que el producto ha sido fabricado originariamente.

El otro matiz es el atributo cantidad. Todo producto que se encuentra dentro del almacén de la empresa tiene una cantidad asignada, es decir, tiene un stock siempre mayor que 0.

3.2.1.3. Clase ProductoFS

La clase ProductoFS modela aquellos componentes que actualmente se encuentran dados de baja en el almacén de la empresa. Esta clase tiene 2 diferencias principales con la explicada anteriormente.

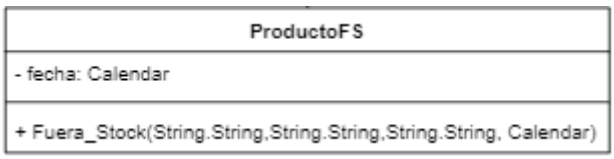

Ilustración 37. Clase ProductoFS

El primer matiz que se puede observar es que, para el producto fuera de stock, se desea guardar la fecha de baja del producto.

Para los productos en stock, la antigüedad del producto viene dada por el albarán del proveedor. Este albarán consta de una serie de números ordenados de menor a mayor, por tanto, el componente es más antiguo cuanto menor sea el número del albarán. Gracias a esto se sigue respetando la gestión FIFO en cuanto a la entrada/salida de los componentes.

Almacenar la fecha de baja para un producto dado de baja es esencial para conservar una trazabilidad en la llegada y salida de distintos componentes y proyectos, además de para respetar las normas de calidad que se exigen en la empresa, debido al cumplimiento de la norma ISO 9001:2015.

3.2.1.4. Clase Ubicación

La clase Ubicación almacena y modela las posiciones de cada producto almacenado en la base de datos. Tanto de los productos que se encuentran actualmente en el almacén, como aquellos productos que ya no se encuentran guardados. Estas ubicaciones se guardan para que, en caso de que vuelvan a llegar componentes iguales, se coloquen en el mismo sitio.

Todo producto, ya sea en stock o fuera de él, debe tener obligatoriamente una ubicación debido a la relación de composición que se obtiene del diagrama de clases. No puede haber productos que no tengan ubicaciones.

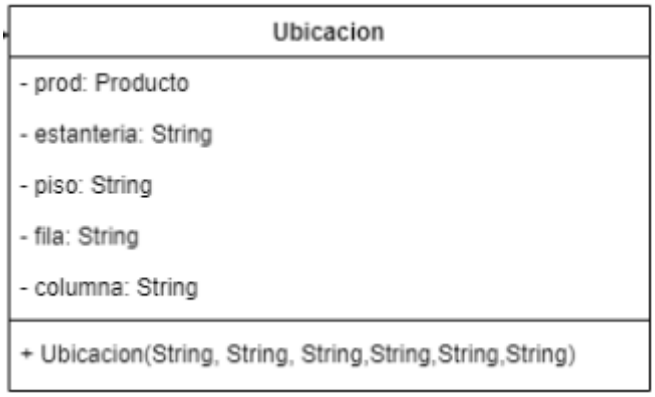

Ilustración 38. Clase Ubicación

3.2.2. Implementación de la capa de Persistencia

La capa de persistencia, como hemos comentado anteriormente, es la que se encarga de manipular la base de datos de la aplicación, conecta el modelo con la lógica de negocio, proporcionándole la funcionalidad a ésta para que la capa final pueda obtener los métodos completos para controlar el sistema.

Para esta implementación se desarrollará una clase que controlará y manipulará la base de datos, implementando todas las operaciones CRUD necesarias para la completa obtención de los objetos.

A esta clase la denominaremos GestorBD.

Esta clase también contiene métodos para la obtención de la conexión a la base de datos de la aplicación, explicaremos en el siguiente apartado cómo se ha desarrollado el tema.

3.2.2.1. Clase GestorBD

En primer lugar, mostraremos el diagrama UML de la clase en cuestión para tratar los diferentes detalles seguidamente:

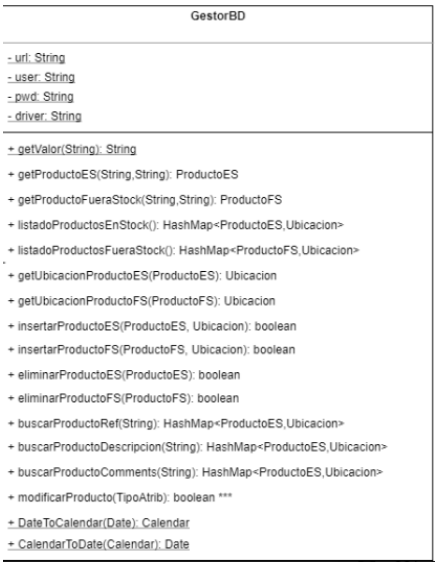

Ilustración 39. Clase GestorBD

Es una clase muy extensa, con diferentes atributos y métodos. A continuación explicaremos los detalles más importantes de esta implementación.

Primeramente, los atributos nunca se utilizan en los métodos propiamente de la base de datos explícitamente. Estos atributos son necesarios para obtener una conexión con la base del sistema, de esta manera se podrán utilizar aquellos métodos “CRUD” para manipular los productos.

Los valores de estos atributos se obtienen de un fichero de propiedades. Este fichero tiene una extensión .properties y debemos leer y almacenar los valores en las variables anteriores para realizar de manera dinámica las conexiones cada vez que se utiliza un método de acceso o manipulación de la base de datos.

Para ello se ha desarrollado un bloque estático para asignar los valores a los atributos y un método para obtener estos valores del fichero de propiedades.

Los atributos son estáticos y se componen de:

- URL: La url consta de la línea de conexión para acceder a la base de datos del sistema, contiene la IP y el puerto por el que conectarse a esta base de datos. Además de otras propiedades de configuración.
- User: El usuario contiene el nombre de acceso a la base de datos. Es decir, el nombre de usuario de acceso a la base de datos, no el acceso al puesto de trabajo.
- Pwd: El pwd contiene la contraseña de acceso a la base de datos. Al igual que el campo user, se trata de la contraseña de acceso a la base de datos, no la contraseña de acceso al puesto de trabajo.
- DriverClass: Contiene el tipo de driver de conexión a la base de datos del sistema. Es muy importante elegir el driver adecuado, si no se selecciona bien, la aplicación no puede funcionar.

```
/**
 *
 * Método de obtención de valores para la conexión a la base de datos
 *
 * @param clave El parámetro clave define el nombre del valor del fichero de
 * propiedades de conexión a la BD.
 *
 * @return devuelve el valor asociado a la clave pasada por parámetro
 *
 */
public static String getValor(String clave) {
    Properties prop = new Properties();
    try {
        prop.load(new FileInputStream("C:\\conexion.properties"));
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    String valor = prop.getProperty(clave);
    return valor;
}
```

Ilustración 40. Método estático de obtención de valores del fichero de conexión.

```
static {
    try {
        url = getValor("url");
        user = getValor("user");
        pwd = getValor("pwd");
        driver = getValor("driverClass");
        Class.forName(driver);
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
}
```

Ilustración 41. Bloque estático de obtención de valores y almacenamiento en las variables

El objetivo de esta porción de código es dinamizar las conexiones a la base de datos de los métodos de manipulación de esta. Es decir, este fragmento de código se debería utilizar en cada uno de los métodos, pero de esta manera se optimiza la escritura de código de programación.

Para el resto de métodos esenciales se mostrará una imagen de cada uno de ellos. En cada operación esencial se desarrolla código similar:

```
public ProductoES getProductoES(String ref, String alb) {
    ProductoES prod = null;
    Connection con = null;
    try {
        con = DriverManager.getConnection(url, user, pwd);
        try (Statement stmt = con.createStatement()) {
            String sql = "SELECT * FROM PRODUCTO_ES WHERE REF_FABRIC = '" + ref + "' AND ALBARAN = '" + alb + "' ";
            try (ResultSet res = stmt.executeQuery(sql)) {
                if (res.next()) {
                    prod = new ProductoES(ref, alb, res.getString(3), res.getString(4),
                            res.getString(5), res.getString(6), res.getString(7), res.getInt(8));
                }
            }
        }

    } catch (NullPointerException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(null, "El producto que se desea manipular no existe", "Error.", JOptionPane.WARNING_MESSAGE);

    } catch (SQLException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(null, "No se ha podido establecer conexión con la base de datos.", "Error de conexión", JOptionPane.ERROR_MESSAGE);

    } finally {
        if (con != null) {
            try {
                con.close();
            } catch (SQLException ex) {
                ex.printStackTrace();
            }
        }
    }
    return prod;
}
```

Ilustración 42. Método de obtención de un Producto

```
public boolean modificarRef(String nuevaref, String viejaref, String albaran) {

    Connection con = null;

    PreparedStatement pstmt1 = null;
    PreparedStatement pstmt2 = null;

    String sql1 = "UPDATE PRODUCTO_ES SET REF_FABRIC = ? WHERE REF_FABRIC = ? AND ALBARAN = ?";
    String sql2 = "UPDATE UBICACION_PRODUCTO SET REF_FABRIC = ? WHERE REF_FABRIC = ? AND ALBARAN = ?";

    int i = 0;

    try {
        con = DriverManager.getConnection(url, user, pwd);
        con.setAutoCommit(false);

        pstmt1 = con.prepareStatement(sql1);
        pstmt1.setString(1, nuevaref);
        pstmt1.setString(2, viejaref);
        pstmt1.setString(3, albaran);
        i = pstmt1.executeUpdate();

        pstmt2 = con.prepareStatement(sql2);
        pstmt2.setString(1, nuevaref);
        pstmt2.setString(2, viejaref);
        pstmt2.setString(3, albaran);
        i = pstmt2.executeUpdate();

        pstmt1.close();
        pstmt2.close();
        con.commit();
        JOptionPane.showMessageDialog(null, "El producto ha sido modificado.", "Producto Modificado", JOptionPane.INFORMATION_MESSAGE);
```

Ilustración 43. Método de Modificación de la Referencia de un producto

En el caso de los listados de productos, tanto el total como las búsquedas filtradas por referencia, descripción y orden de trabajo, se han usado HashMaps para el almacenamiento de estos datos. Esto se debe a que, tanto el producto como su ubicación son objetos que, de alguna manera, deben estar unidos. Es por eso que se usa esta estructura teniendo como clave la referencia del objeto, y como valor la ubicación.

Como apunte final, debemos comentar el asunto del tratamiento de fechas:

- En Java, las fechas en este caso se guardan como objetos de tipo Calendar. Es cierto que, también existe un tipo Date, pero Calendar da más juego en cuanto a los formatos y las posibilidades de manipulación que ofrecen.
- En el caso de SQL, únicamente trata con objetos del tipo Date de sus librerías.
- Para conectar estos dos tipos de objeto se han desarrollado dos métodos estáticos de conversión de fechas, es decir, de paso de Calendar a Date y viceversa.

```java
/**
 *
 * Método de conversión de un objeto de tipo Date a un objeto del tipo
 * Calendar, esencial para trabajar con las fechas en la BD
 *
 *
 * @param fecha El parametro fecha indica la fecha que se desea convertir
 *
 * @return Devuelve la fecha pasada por parametro en formato Calendar
 *
 */
public static Calendar DateToCalendar(Date fecha) {
    Calendar c = Calendar.getInstance();
    c.setTime(fecha);
    return c;
}

/**
 *
 * Método de conversión de un objeto de tipo Calendar a un objeto del tipo
 * Date, esencial para trabajar con las fechas en la BD
 *
 *
 * @param fecha El parametro fecha indica la fecha que se desea convertir
 *
 * @return Devuelve la fecha pasada por parametro en formato Date
 *
 */
public static Date calendarToDate(Calendar fecha) {
    Date f = new Date(fecha.getTimeInMillis());
    return f;
}
```

Ilustración 44. Métodos de conversión de fechas

3.2.3. Implementación de la Lógica de Negocio

La capa de lógica de negocio se encarga de conectar la base de datos con el nivel de usuario de la aplicación o la capa de presentación. Es decir, esta capa hace de "puente" entre los datos a utilizar por el usuario y la funcionalidad que este posee para manejarlos.

Para esta implementación se desarrollará una clase que usará los métodos que hay implementados en la capa de persistencia, aquellos que están en la clase GestorBD, para que las clases de la capa de presentación los utilicen.

A esta clase la denominaremos LogicaNegocio.

3.2.3.1. Clase LogicaNegocio

Como hemos dicho en el apartado anterior, esta clase sirve de puente entre las clases GestorBD y capa de presentación. La implementación de esta clase es simple, usando únicamente los métodos de la clase correspondiente a la manipulación de la base de datos bajo unas condiciones, en general, de nulidad.

Estas condiciones de nulidad se basan en que los métodos no tengan que ejecutarse cuando aquello que se pide para la ejecución del método sea nulo o no exista en la base de datos.

3.2.4. Implementación de la capa de Presentación

La capa de presentación es el nivel de abstracción final. Es el conjunto de funcionalidades que el usuario ve y le permite interactuar con la base de datos de la aplicación mediante interfaces gráficas.

Para esta implementación se desarrollará una clase que, además de generar esas interfaces gráficas, se incluye la funcionalidad de los botones que realizan las funciones para el usuario final del sistema

A esta clase se le llamará Interfaz.

3.2.4.1. Clase Interfaz

Esta clase se encarga de la totalidad de la capa de presentación, tanto de las interfaces gráficas como de proporcionar la funcionalidad del sistema a éstas.

En primer lugar vamos a explicar la interfaz gráfica y después cada una de sus funcionalidades:

3.2.4.1.1. Parte gráfica

La interfaz gráfica es única y sencilla, se basa en un título principal con la imagen corporativa de la empresa, para a continuación desplegar un menú de 6 pestañas, donde cada una posee un contenedor con la funcionalidad de cada una de las operaciones que se desean realizar: Búsqueda, inserción, modificar, eliminar, listado completo y stocks de productos.

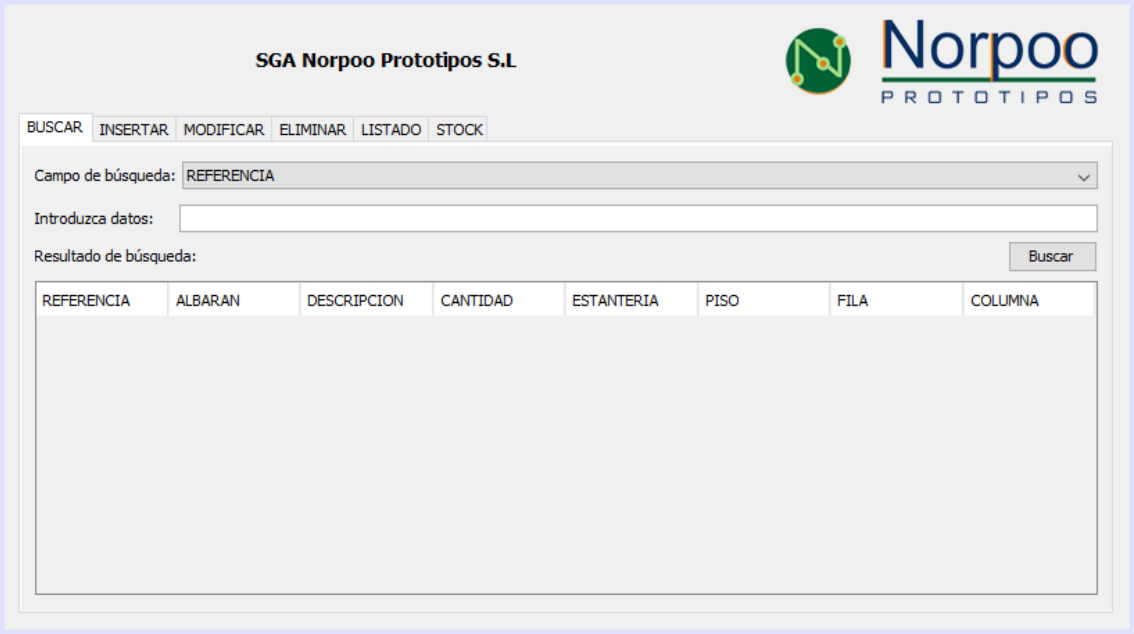

Ilustración 45. Interfaz gráfica de la Aplicación

Para el método de búsqueda se incluyen varios objetos como un ComboBox para seleccionar el criterio de búsqueda, un TextBox para introducir los datos que se desean seleccionar, un botón para ejecutar la búsqueda y una JTable para mostrar los resultados. Este número de objetos es similar en la interfaz perteneciente al Stock y al Listado.

En los apartados de la modificación, inserción y eliminación de objetos, la interfaz se basa en la introducción de los datos necesarios para realizar la operación así como un botón para ejecutarla.

3.2.4.1.2. Parte de desarrollo del código

En cuanto a la parte de desarrollo de código, aquellos componentes que se colocan en la interfaz gráfica se generan en el código de manera automática. En este apartado se va a explicar la funcionalidad de los botones que el usuario puede clicar y ejecutar las operaciones de manipulación de los datos.

Para los métodos tanto de búsqueda, como de listado total y stocks de producto disponible se ha seguido el mismo patrón de desarrollo.

En primera instancia se toman los valores que hagan falta para realizar la búsqueda si es necesario almacenar algún valor. Tras esto, se genera la tabla realizando un TableModel y se coloca una limpieza de filas de tabla para que, cada vez que se ejecute una acción no se acumulen las filas de los resultados de la búsqueda o los listados.

Tras esto, las entradas del HashMap de los resultados se almacenan en una lista de Entradas, esto es debido a que las listas se pueden ordenar por un criterio de manera ascendente o descendente. Este orden es esencial porque el HashMap muestra los datos desordenados y cada ejecución enseña un orden diferente.

Esta colección se ordena por la clave del HashMap, es decir, la referencia del producto para que la muestra del listado aparezca ordenada por ese criterio.

```
private void BT_stockActionPerformed(java.awt.event.ActionEvent evt) {

    String referencia = this.TF_stock_ref.getText();
    DefaultTableModel dtm = (DefaultTableModel) tablastock.getModel();
    dtm.setRowCount(0);
    Map<ProductoES, Ubicacion> prods = LogicaNegocio.BUSCARproductoREF(referencia);
    List<Map.Entry<ProductoES, Ubicacion>> listaEntradas = new ArrayList<>(prods.entrySet());
    if (prods.isEmpty()) {
        JOptionPane.showMessageDialog(null, "No hay stock con esa referencia.", "Error al mostrar productos.", JOptionPane.WARNING_MESSAGE);
    } else {

        Collections.sort(listaEntradas, (Map.Entry<ProductoES, Ubicacion> entrada1, Map.Entry<ProductoES, Ubicacion> entrada2) -> entrada1.getKey().getRef_

        for (int i = 0; i < listaEntradas.size(); i++) {
            dtm.addRow(new Object[]{listaEntradas.get(i).getKey().getRef_fabric(), listaEntradas.get(i).getKey().getAlbaran(), listaEntradas.get(i).getKey
        }
    }

}
```

Ilustración 46. Método de Stock de productos disponibles

Los métodos de modificación e inserción siguen un patrón parecido, se basan en un conjunto de TextBox donde se introducen los valores que sean necesarios para la inserción (todos los campos del producto y su ubicación) o la eliminación (referencia y albarán). En el desarrollo del método se realizan las comparaciones y condicionales para la correcta funcionalidad del método. Finalmente, un cuadro de diálogo aparece para decidir si se desea continuar. En caso afirmativo se realiza la acción y en caso contrario se para y se vuelve a la ventana principal.

```
String referencia = this.TF_insertar_ref.getText();
if ("".equals(referencia)) {
    JOptionPane.showMessageDialog(null, "Tienes que introducir una referencia, si no no puedes insertar", "Inserta una referencia.", JOpt
    return;
}
String albaran = this.TF_insertar_albaran.getText();
if ("".equals(albaran)) {
    JOptionPane.showMessageDialog(null, "Tienes que introducir un albaran, si no no puedes insertar", "Inserta un albaran.", JOptionPane.
    return;
}
String proveedor = this.TF_insertar_prov.getText();
if ("".equals(proveedor)) {
    proveedor = null;
}
String datacode = this.TF_insertar_datacode.getText();
if ("".equals(datacode)) {
    datacode = null;
}
String descripcion = this.TA_insertar_desc.getText();
if ("".equals(descripcion)) {
    descripcion = null;
}
String hum = this.TF_insertar_hum.getText();
if ("".equals(hum)) {
    hum = null;
}
if ("S".equals(hum)) {
    JOptionPane.showMessageDialog(null, "Al indicar S en humedad, el producto debe ir en la estanteria AD101", "Estanteria de humedad.",
    this.TF_insertar_estanteria.setText("AD101");
}
String coms = this.TF_insertar_com.getText();
if ("".equals(coms)) {
    coms = null;
}
String cantidad = this.TF_insertar_cant.getText();
if ("".equals(cantidad)) {
    cantidad = "0";
}
```

Ilustración 47. Extracto de método de inserción de productos

El método de eliminación es parecido a los métodos de inserción o modificación con un detalle a comentar. El procedimiento es igual a la modificación, pero, si el producto a eliminar existe en la base de datos, antes de eliminarlo se inserta en la tabla correspondiente a productos fuera de stock. Cuando se inserta en esta tabla y se almacenan los datos necesarios, el producto se elimina.

```
private void BT_eliminarActionPerformed(java.awt.event.ActionEvent evt) {
    String referencia = this.TF_eliminar_ref.getText();
    String albaran = this.TF_eliminar_albaran.getText();
    ProductoES p = LogicaNegocio.GETproductoES(referencia, albaran);
    Ubicacion u = LogicaNegocio.GETubicacionProductoES(p);
    Calendar fecha = Calendar.getInstance();
    if (p != null || u != null) {
        int confirmar = JOptionPane.showConfirmDialog(null, "¿Estás seguro de querer eliminar el producto?", "Atención!", JOptionPane.YES_NO_OPTION);
        if (JOptionPane.YES_OPTION == confirmar) {
            ProductoFS fs = new ProductoFS(p.getRef_fabric(), p.getAlbaran(), p.getProveedor(), p.getDescripcion(), p.getHum(), p.getComentarios(), fecha)
            LogicaNegocio.INSERTARproductoFS(fs, u);
            LogicaNegocio.ELIMINARproductoES(p);
        } else {
            JOptionPane.showMessageDialog(null, "No se ha encontrado el producto a eliminar", "Error al eliminar.", JOptionPane.WARNING_MESSAGE);

        }
    }

}
```

Ilustración 48. Método de eliminación de un producto + Inserción en histórico de productos

4. Etiquetado de productos

Una vez implementado y comprobado que el programa, tanto su funcionalidad a nivel de usuario, como a nivel de base de datos, funciona correctamente, debemos proceder a generar una etiqueta para cada producto a insertar.

Para este apartado se dispone de:

- Impresora Brother QL-700: Pequeña impresora de etiquetas para la impresión de estas.
- Software P-Touch Editor: Programa incluido con la impresora que se encarga del diseño y generación de las etiquetas.

Explicaremos cada paso del etiquetado con detalle.

- En primer lugar, vamos a conectar el software de la impresora de etiquetas con la base de datos del almacén de la empresa. Antes de crear un diseño de etiqueta es necesario realizar la conexión. No se podrá editar la información de la base de datos desde este programa.

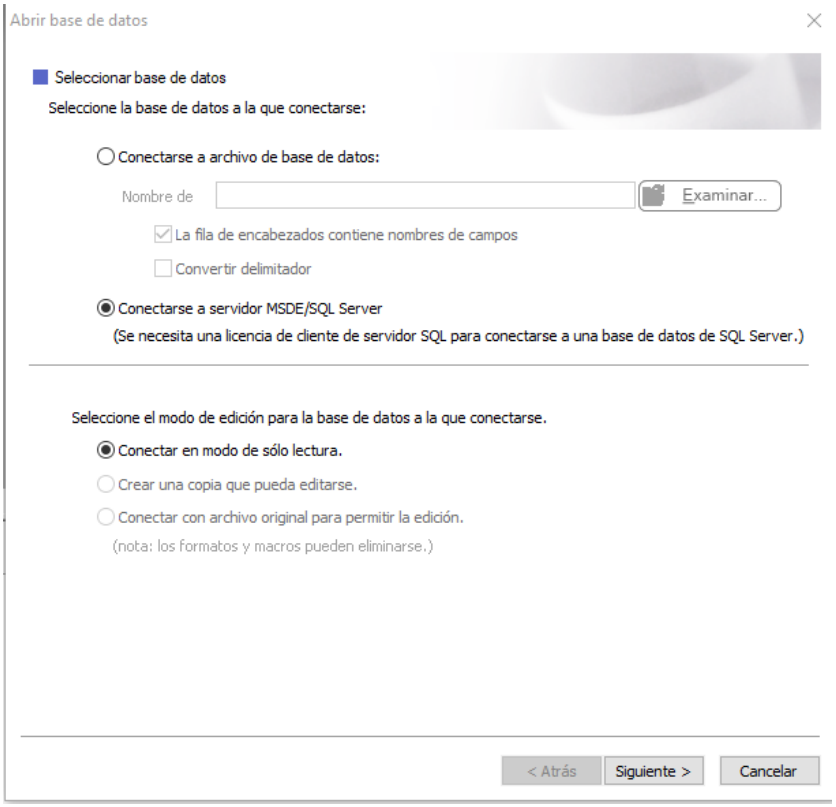

Ilustración 49. Primer paso conexión

- El paso siguiente es proporcionar las credenciales del usuario de la base de datos, en este caso usaremos los del usuario Almacen. Pero cualquier usuario de la empresa puede utilizar el software de etiquetas para imprimir.

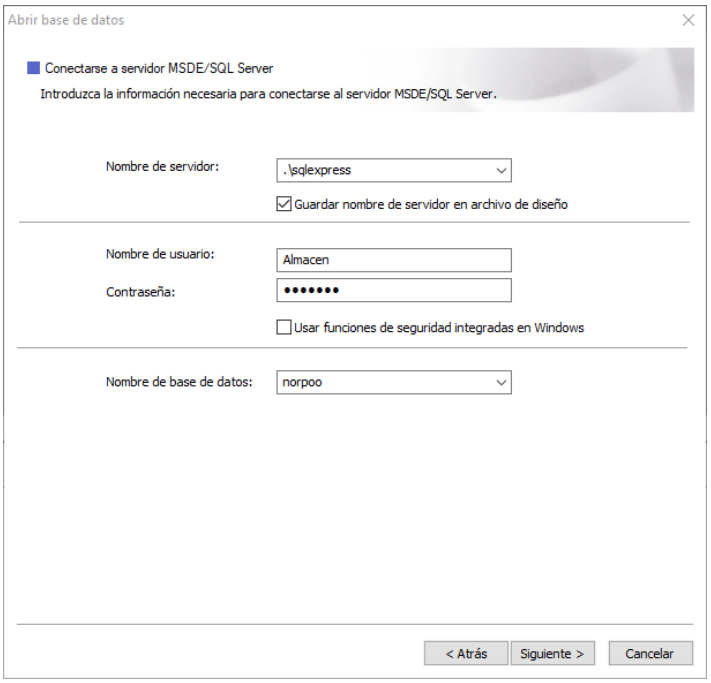

Ilustración 50. Introducción de credenciales para la conexión a SQL Server

Este software tiene la particularidad de que solo podemos seleccionar una tabla de datos para diseñar e imprimir etiquetas, es por esto que se ha creado una vista de datos en el SGBD para su uso exclusivo en la parte del etiquetado. Esta vista compone un JOIN de las tablas PRODUCTO_ES y UBICACIÓN_PRODUCTO, concatenando los campos de la ubicación en uno solo, este aspecto es una ayuda puramente visual para la etiqueta.

La etiqueta va a ser igual para todos los productos, al conectar la base de datos se pueden incluir en la etiqueta objetos dinámicos. Estos objetos dinámicos son los campos de la vista. El diseño de la etiqueta de un producto es el que se muestra en la Ilustración 51.

Ilustración 51. Etiqueta de producto

Los códigos especiales contienen valores identificativos del producto, es decir, la referencia y el albarán en los códigos QR, y la ubicación del producto en el código de barras.

Se prevé que en un futuro estos códigos tengan especial importancia debido a la adquisición de diferente material para automatizar procesos.

Tras el diseño de la etiqueta, procedemos a imprimir las necesarias para identificar todo el almacén.

Ilustración 52. Producto etiquetado listo para su almacenaje

5. Consideraciones sobre el despliegue

En el apartado de análisis y planificación se ha explicado la estructura de red de ordenadores de la empresa, como los diferentes roles de usuarios que existen en ella. La aplicación de escritorio será desplegada en todos los equipos, compartiendo un acceso directo a cada puesto de trabajo. Todos los archivos necesarios para su funcionamiento se encontrarán en una carpeta compartida utilizada para tener diferente información interna en todos los puestos.

En cada puesto, como hemos comentado anteriormente en la fase de implementación, se desarrollará un fichero de propiedades de conexión a la base de datos. Estas propiedades constarán del driver de conexión, la url de esta, usuario de acceso a la base de datos y contraseña de acceso a la base de datos.

La base de datos estará ubicada en el puesto de trabajo con más permisos en la base de datos, es decir, el ordenador perteneciente al empleado del almacén. Todos los demás puestos se conectarán remotamente a este puesto, que actuará como servidor de base de datos.

Realizado todo este protocolo, la aplicación estará disponible para su uso desde todos los ordenadores de la empresa.

6. Otros aspectos y conclusiones

6.1. Manual de usuario

Tras completar todo el proceso de creación, diseño y desarrollo del proyecto, se ha elaborado un manual de usuario para la familiarización con el programa. Este manual se convertirá en documentación interna de la empresa, accesible para todos los empleados.

El manual se encontrará en los Anexos del Proyecto.

6.2. Mantenimiento

El sistema se revisará periódicamente para la optimización del código o diferentes cuestiones en cuanto al rendimiento. Cuando suceda algún cambio este se implementará y se realizará el despliegue de la nueva versión a todos los equipos de la empresa.

Además, si la empresa en cuestión desea incluir alguna funcionalidad más, esta se valorará en tiempo y seguirá el mismo proceso que tuvieron aquellos servicios que ya están implementados.

6.3. Otras conclusiones

Gracias a este proyecto la trazabilidad de los componentes que se encuentran en el almacén de la empresa Norpoo Prototipos SL se ha visto mejorada notablemente. Siendo esto necesario para cumplir con las exigencias de calidad que propone la norma ISO:9001, conseguida también en el proceso de prácticas externas en Diciembre de 2018.

Este desarrollo de proyecto se ha utilizado para la generación de otros programas de utilidad para la empresa, como un gestor interno de los mantenimientos de las máquinas de producción. También necesidad de empresa para el cumplimiento de los estándares de calidad.

Por último, agradecer a la empresa Norpoo Prototipos su confianza en mis servicios además de otorgarme la responsabilidad del desarrollo de esta necesidad tan importante para la empresa y espero seguir colaborando y trabajando en futuros proyectos. Y agradecer a la Universidad de La Rioja porque, sin el servicio de prácticas externas, este trabajo no hubiese sido posible realizarlo.

7. Bibliografía

- Normativa ISO 9001:2015. (Proporcionada por la empresa)
- Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK), Quinta Edición. 2013
- https://www.omg.org/spec/UML/2.5.1/PDF - Guía de UML para consulta de diagramas
- https://docs.microsoft.com/en-us/sql/connect/jdbc/step-3-proof-of-concept-connecting-to-sql-using-java?view=sql-server-2017
- https://stackoverflow.com/questions/40471/differences-between-hashmap-and-hashtable
- https://docs.oracle.com/javase/7/docs/api/java/util/Hashtable.html
- https://docs.oracle.com/javase/8/docs/api/java/util/HashMap.html
- https://docs.microsoft.com/es-es/sql/connect/jdbc/connection-url-sample?view=sql-server-ver15
- https://stackoverflow.com/questions/3922337/how-to-create-composite-primary-key-in-sql-server-2008/3922359
- https://docs.oracle.com/javase/7/docs/api/javax/swing/JTable.html

- https://docs.oracle.com/javase/8/docs/api/java/util/Collections.html
- https://stackoverflow.com/questions/91688/what-are-the-differences-between-a-clustered-and-a-non-clustered-index#:~:text=Clustered%20indexes%20are%20stored%20physically,unique%20column%2C%20usually%20the%20PK.